QUALITY OF SERVICE (QOS) SECURITY IN MOBILE AD HOC NETWORKS

A Dissertation

by

BIN LU

Submitted to Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

August 2005

Major Subject: Computer Science

QUALITY OF SERVICE (QOS) SECURITY IN MOBILE AD HOC NETWORKS

A Dissertation

by

BIN LU

Submitted to Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Approved by:

Chair of Committee,	Udo W. Pooch
Committee Members,	Riccardo Bettati
	Jianer Chen
	Michael T. Longnecker
Head of Department,	Valerie E. Taylor

August 2005

Major Subject: Computer Science

ABSTRACT

Quality of Service (QoS) Security in Mobile Ad Hoc Networks.
(August 2005)
Bin Lu, B.S.; M.S., Harbin Institute of Technology
Chair of Advisory Committee: Dr. Udo W. Pooch

With the rapid proliferation of wireless networks and mobile computing applications, Quality of Service (QoS) for mobile ad hoc networks (MANETs) has received increased attention. Security is a critical aspect of QoS provisioning in the MANET environment. Without protection from a security mechanism, attacks on QoS signaling system could result in QoS routing malfunction, interference of resource reservation, or even failure of QoS provision.

Due to the characteristics of the MANETs, such as rapid topology change and limited communication and computation capacity, the conventional security measures cannot be applied and new security techniques are necessary. However, little research has been done on this topic. In this dissertation, the security issues will be addressed for MANET QoS systems.

The major contributions of this research are: (a) design of an authentication mechanism for ad hoc networks; (b) design of a security mechanism to prevent and detect attacks on the QoS signaling system; (c) design of an intrusion detection mechanism for bandwidth reservation to detect QoS attacks and Denial of Service (DoS) attacks. These three mechanisms are evaluated through simulation.

DEDICATION

To my parents and my husband

ACKNOWLEDGMENTS

There are many people who contributed to this dissertation in many ways.

First, I thank my advisor, Dr. Pooch, for his many years of patience, support and advice in this work and other areas of academic life. He fostered a stress-free working relationship, which was crucial to the completion of this work.

I would also like to thank my committee members, Dr. Bettati, Dr. Chen, and Dr. Longnecker, for their guidance and support through the course of this research.

The writing of a dissertation can be a lonely and isolating experience, yet it is obviously not possible without the personal and practical support of numerous people. Thus, sincere gratitude also goes to my parents, my husband and my friends.

I would like to thank my parents for their guidance and support in setting my life goals, and for their faith in my achieving these goals.

My husband, Mr. Hengzhi Ai, deserves my deepest thanks and respect for his continued support during the writing of this dissertation. I could not have done it without his love and support.

TABLE OF CONTENTS

Page

Page

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

Page

1. INTRODUCTION

With the rapid proliferation of wireless networks and mobile computing applications, providing Quality of Service (QoS) in an efficient and scalable manner in mobile ad hoc networks (MANETs) has become a topic of active research.

MANETs are characterized by the absence of fixed infrastructure, rapid topology change and high node mobility. These characteristics can be used to determine that the *guaranteed QoS* proposed in wired networks cannot be directly applied to wireless ad hoc networks, because the communication capacity between any two nodes can be dramatically changed and this could result in breaking the previously promised QoS. Instead, *soft QoS* is provided in mobile ad hoc networks. Namely, each node in MANETs can provide only a promise not to deliberately oversubscribe itself and not to intentionally prevent resources from being available.

Security is a critical issue and offers serious challenges in QoS provisioning in wireless ad hoc networks, and yet there is little work published in this area.

Security mechanisms are utilized to preserve protected information and network resources, therefore can protect QoS from being tampered with by adversaries. The security properties that should be supported in MANET QoS include *availability*, *authenticity*, *integrity* and *confidentiality*. *Availability* refers to the requirement that the service offered by the node should be available to its users when expected. It is a primary security property ensuring soft QoS provision. *Authenticity* ensures the principals with whom one interacts are the expected nodes. *Integrity* enforces that a node or message transmitted has not been maliciously altered and *confidentiality* protects the secrecy of communication.

This dissertation follows the style of *IEEE Transactions on Computers*.

Malicious attacks on MANET QoS could target any and all of the above security properties and could be in forms of theft of service or denial of service (DoS), IP address spoofing, malicious corruption or alteration of packets, eavesdropping, etc.

1.1. Why Security for QoS?

The characteristics of ad hoc networks such as exposure to hostile environment (e.g. battle field, rescue missions) and difficulty of authentication exacerbate the QoS model security problems. Without protection of security mechanisms, a QoS model is vulnerable to both theft of service and denial of service, which inhibits the guarantee of network resource availability.

A QoS model specifies an architecture in which some kinds of services could be provided. The objective is to implement a scalable, flexible and secure QoS model. Up to date, *Integrated Services* (IntServ) [1] and *Differentiated Services* (DiffServ or DS) [2] have been proposed to support QoS in the traditional Internet and are also being studied for MANET environments.

The IntServ model provides an end-to-end QoS guarantee on a per-flow basis. It requires that every IntServ-enabled router keep the flow-specific states including bandwidth requirements, delay bound and cost of the flow, and therefore is not scalable for the Internet. DiffServ model is designed to overcome the scalability problem in the IntServ for wired networks. The DiffServ model is based on flow aggregation by classifying packets into a limited number of classes and then applying specific forwarding treatment to each QoS class.

Flexible QoS Model for MANETs (FQMM) [3] is a model proposed solely for mobile ad hoc networks. The FQMM takes the characteristics of MANETs into account and is a

hybrid provisioning scheme of the per-flow service in IntServ and the per-class service in DiffServ.

Although Diffserv model provides more scalability and greater flexibility than the Intserv model, several vulnerabilities in DiffServ for MANETs make it a less secure model than the IntServ.

This research attempts to design a security system to protect the IntServ architecture for mobile ad hoc networks.

Targeting IntServ model in MANETs, adversaries could issue attacks in the following ways:

- A malicious node can tamper QoS provision with falsified data or QoS signaling messages to steal or deplete resources used or reserved by other nodes.

- Attacks on QoS signaling system such as malicious alteration of the QoS parameters in QoS signaling messages. This form of attack could result in incorrect QoS reservation along a path and therefore lead to degradation of network resources utilization or legitimate traffic penalization.

- Advertisement of false network resource information. In MANETs, the network resource information is inaccurate. However, deliberately advertising false information is more dangerous because it will result in incorrect routing and QoS reservation and thus also degradation of network resources utilization or legitimate traffic penalization.

- Maliciously drop, delay or corrupt data packets, resulting in deliberately violating promised QoS.

Therefore, security mechanisms are needed to prevent QoS systems from being maliciously attacked.

1.2. Contributions of This Research

The objective of this research is to provide security protection and intrusion detection mechanisms to prevent from or to detect malicious attacks. We concentrate on authentication approaches, secure QoS signaling systems, and intrusion detection for bandwidth reservation in MANETs. Considering these goals of MANET QoS security, the contributions of this research include:

1) Design a lightweight authentication protocol that can provide integrity and authenticity to neighboring communications in QoS-enabled networks.

Most ad hoc networks do not employ any network access control, leaving them vulnerable to resource consumption attacks. In QoS-enabled ad hoc networks, users need to assure that the party who sent a message is indeed the legitimate party. Otherwise, a malicious node can tamper a network and QoS provision with falsified data and QoS signaling messages to steal or deplete resources used or reserved by other nodes. To deal with these attacks, an authentication protocol needs to be in place to ensure that a packet is sent by an authentic and legitimate node.

In this dissertation, we will propose a lightweight authentication protocol that effectively and efficiently provides security properties such as authenticity and integrity for communicating nodes in MANETs. The protocol not only eliminates the high performance overhead imposed by *asymmetric* cryptography (such as digital signatures), but also avoids the difficulty of key management introduced by secret paired *symmetric* key. The authentication protocol is proved to be lightweight, scalable and tolerant of packet loss.

2) Build an Intrusion Prevention mechanism and an Intrusion Detection System (IDS) to prevent and detect attacks on QoS signaling.

The vulnerabilities and types of security violations will be analyzed for MANET QoS models, which include IntServ model, Diffserv model and the Flexible QoS Model for MANETs. The analysis demonstrated that DiffServ and FQMM are vulnerable to attacks such as theft and depletion of network resources. Compared to the DiffServ model, the IntServ approach does not have the security risks mentioned above because it is based on flows rather than on aggregated traffic as in DiffServ and FQMM. However, IntServ model requires a signaling system to achieve QoS provision along a data path.

QoS signaling is used to search for routes with sufficient resources for desired QoS, to reserve and release resources, to set up, tear down and renegotiate flows. Without protection from a security mechanism, attacks on QoS signaling system could result in QoS routing malfunction, interference of resource reservation, or even failure of QoS provision. Current approaches proposed for intrusion detection and security prevention on QoS signaling in wired networks (such as SDS/CD and RSVP-SQOS) cannot be applied to ad hoc QoS signaling systems due to various reasons.

Part of this research is to develop an intrusion prevention mechanism as well as a set of rules that can be used to effectively and efficiently detect attacks on QoS signaling (bandwidth, delay or jitter parameters) in mobile ad hoc networks.

This mechanism is aimed at meeting the following security *requirements*:

- QoS parameters delivered in signaling messages can be classified as *non-mutable* parameters (such as requested bandwidth, delay or jitter) and *mutable* parameters (such as those used to measure available bandwidth or delay along the path). An integrity protection mechanism should be in place to guarantee

that the non-mutable part in the QoS object, such as the QoS profiles for traffic flows, is not changed illegally.

- QoS states collected over the path (e.g. available bandwidth and accumulative delay over a path) should be resistant to attacks, which are stored in *mutable* QoS parameters. The malicious attacks on these parameters are more deceiving than those on non-mutable ones because they cannot be detected via integrity verification.

3) Build an Intrusion Detection System to detect attacks on bandwidth reservation in mobile ad hoc networks.

In the traditional network, once resources are successfully reserved along a path, the QoS is expected to be guaranteed. Breaking the bandwidth reservation is unusual and thus can be considered as presence of malicious attacks. The case is different in ad hoc networks because there is only *soft QoS*, which means a node will not intentionally or knowingly oversubscribe itself to make the resource unavailable to the traffic after resource reservation. Therefore the QoS is not guaranteed but only promised and nodes are allowed to break the promise in case of abrupt resource changes due to mobility, wireless interference or the node being shut down. Adversaries could take advantage of this characteristic and issue an attack by means of pretending to reserve the resource while break the promise afterwards. It can lead to excessive overhead to the traffic and degradation of network performance.

In MANETs, a break of QoS promise can result from malicious attacks as well as radio interference from the nodes who just “wandered” into the neighborhood unaware of the reservation. Moreover, communication links in MANETs are open medium and therefore subject to radio interference. Detection of intrusion on bandwidth reservation needs to distinguish these cases and apparently is not a trivial task.

An algorithm is proposed in this dissertation to detect both DoS attacks (issued by malicious nodes in the neighborhood to disrupt the service), and QoS attacks (issued by relay node on the path to disrupt the service or to steal the bandwidth).

The performance of the proposed security mechanisms are evaluated through simulation.

1.3. Outline

This dissertation is organized as follows: Chapter II describes the related work in QoS issues and security issues in mobile networks. In this chapter, we also give a brief introduction on QoS security problems in both the traditional Internet and wireless networks. In Chapter III, we will propose a lightweight authentication protocol that can be used to protect neighboring communications in mobile ad hoc networks. We first describe the assumptions used for the design. Then we will introduce the trust management and the message authentication schemes. The security properties and the performance evaluation from simulation will be demonstrated too. This authentication protocol will be used to protect the authentication and integrity of QoS signaling system which will be addressed in Chapter IV. Chapter IV studies the security problems for QoS systems in MANETs. The security vulnerabilities in each QoS model (such as IntServ, DiffServ and FQMM) will be analyzed first. Secondly, we will propose a security mechanism for QoS signaling system. Simulation results will be showed at last. In Chapter V, we will propose an intrusion detection mechanism for bandwidth reservation in MANETs. The attack models for bandwidth reservation will be described and then the detection scheme will be introduced. Chapter VI will conclude the work and discuss about future work.

2. RELATED WORK

The characteristics of MANETs determine that providing QoS in MANETs is different from that in the traditional Internet. Each node in MANETs can provide only a promise not to deliberately oversubscribe itself and not to intentionally prevent resources from being available [4], which introduces difficulty in providing security properties to QoS in MANETs.

The security properties that should be supported in MANET QoS include *availability*, *authenticity*, *integrity* and *confidentiality* [5]. More specifically, the QoS security problems to be solved are as follows: [6]

- Protection of crucial Quality of Application (QoA) parameters during connection setup. The protection is at the *control level*.
- Protection of data packets during their transmission in a timely manner. This protection is at the *data level*.
- Protection against denial of service attacks.

Authentication, access control, encryption, denial-of-access-sensitive admission control should be enforced during the QoS connection setup to distribute the QoS requirements and provide proper resource reservation, allocation and access in a secure fashion. If security mechanisms and policies at routers, gateways and firewalls, such as intrusion detection, digital signature and encryption with variable key lengths, scalable key management, watermarking, security policy management are available, this could provide for a secure transmission path, content protection and end-to-end QoS provision.

Up to date, a great amount of research has been done in the study of QoS in MANETs, security in MANETs and QoS security in conventional networks. However, as of our knowledge, there is yet little work published on the topic of QoS security for MANETs.

2.1. QoS in Mobile Ad Hoc Networks

The characteristics of Quality of Service in MANETs significantly affect the QoS architecture and routing protocols employed in MANETs, which are as follows:

- There is no core and edge distinction. All the nodes are homogenous in QoS provision roles. Due to the fact of mobility and absence of a fixed infrastructure, a node can serve as a core node at one time and an edge node at another time.
- The link between two nodes is a shared medium instead of a point-to-point link as in wired networks. Because of the open medium feature, a node in a mobile ad hoc network can have interference from the neighboring nodes in packet transmission.
- High node mobility results in guaranteed QoS proposed in wired networks not applicable to MANET QoS any more. Previously promised QoS can be broken when the communication capacity between two nodes dramatically change. Therefore, each node in MANETs can provide only a promise not to deliberately oversubscribe itself and not to intentionally prevent resources from being available [4].
- The communication capacity between nodes is low. The link bandwidth is within the range of 1Mbps to 11Mbps, which is less than that in the traditional networks. However, the scalability problem of the Internet IntServ model is less likely to occur in the current MANETs in consideration of the small number of traffic flows and the limited size of the network [7].

2.1.1. QoS signaling systems in mobile ad hoc networks

The IntServ model provides an end-to-end QoS guarantee on a per-flow basis. RSVP is a signaling protocol for resource reservation in IntServ model for wired networks, which allow some users to access to preferential networking resources . Permission to make a

reservation will depend both on the availability of the requested resources along the path of the data, and on satisfaction of policy rules.

The approach of making advance reservations to obtain a quality of service that is not affected by mobility for a mobile host is employed along the data flow paths to and from the locations it may visit during the lifetime of the connection. The mobile host can be a sender in a flow, a receiver in a flow or both sender and receiver in the same flow simultaneously. Other than these, the reservation model of RSVP is used.

Two approaches have been proposed to solve the mobile RSVP problem: Mobile RSVP (MRSVP) [8] and Hierarchical Mobile RSVP (HMRSVP) [9].

Both protocols employ *active* and *passive* reservations. For a mobile sender, it makes an active reservation from the current location of the mobile host and it makes passive reservations from the other locations listed in its MSPEC. To improve the utilization of mobile links, the bandwidth with passive reservations of a flow can be used by other flows requesting weaker QoS or best-effort services until the passive reservations become active. Two approaches were proposed to handle the active and passive messages in MRSVP. In the first approach, the proxy agents play a more important role in processing active and passive messages and no other nodes besides the proxy agents and mobile hosts are involved in the RSVP message processing and forwarding rules. The second approach uses some additional objects in RSVP message and extends the RSVP processing and forwarding rules at all nodes, which however ensures better utilization of network resources.

HMRSVP is based on MRSVP but HMRSVP makes advance resource reservations only when an inter-region movement may possibly happen. The simulation models and results are also demonstrated to show that HMRSVP can achieve the same QoS

guarantees as MRSVP with fewer resource reservations.

INSIGNIA signaling system is an in-band signaling system specifically designed to deliver adaptive real-time service in MANETs [10]. The term “in-band signaling” refers to the fact that control information is carried along with data flows; while “out-of-band signaling” indicates the control information is carried in separate control packets and channels distinct from the data path. Based on the in-band approach, the INSIGNIA can restore a reservation in response to topology changes within the interval of two consecutive IP packets under ideal conditions. INSIGNIA performance relies on the speed at which the routing protocol can recompute new routes if no alternative route is cached after topology changes. In the ideal case where cached alternative routes are available, restoration of resource reservation can be made as quickly as the period between two consecutive packets associated with a session as long as sufficient resources are available along the new path. In contrast, out-of-band signaling systems, for example, would need to maintain source route information and respond to topology changes by directly signaling intermediate routers on an old path to allocate/free radio resources. This is impossible in many cases if the affected router is out of radio contact from the signaling entity that attempts to de-allocate resources over the old path. Therefore, INSIGNIA is a lightweight signaling system in terms of the amount of bandwidth consumed for network control and capable of fast flow reservation, restoration and adaptation

Charles Perkins et. al. proposed a QoS signaling system for Ad hoc On-demand Distance Vector (AODV) routing in MANETs [11]. In their draft, extensions are added to the route discovery packets, specifying the service requirements that must be met by nodes re-broadcasting a Route Request (RREQ) or returning a Route Reply (RREP) for a destination. Specifically, a *QoS Object* extension provides the QoS flow profile, including delay and jitter parameters. A *Maximum Permissible Delay (or Jitter)* extension is also included in AODV in order to enable accumulated measurement for

end-to-end delay (or jitter). An intermediate node will generated an *ICMP QOS_LOST* message if it experiences a significant change in its ability to keep the QoS promises it has made to the source of the flows.

D. A. Maltz argued that the Dynamic Source Routing (DSR) protocol should be used for resource reservations for mobile ad hoc networks because his simulation studies showed that DSR had "by far the lowest overhead in terms of routing packets sent among the current set of routing protocols for ad hoc networks" [4]. The two mechanisms – *path-state* and *flow-state* are used to explicitly manage resources in ad hoc networks. *Path-state* allows intermediate nodes to forward packets according to a predetermined source route. The originator of each data packet initially includes both a source route and a unique *path identifier* for the route in each packet it sends. As intermediate nodes forward the packet, they cache the source route from the packet and record the according path identifier. Then the source can send subsequent packets carrying only the path identifier, and intermediate nodes forward the packet based on the source route for the path indexed by the path identifier that they have cached. *Flow-state* allows a source to differentiate its traffic into flows, and therefore to request better-than-best-effort handling for these flows. With the additional information provided by the flow-state, the network will be able to provide admission control and promise a specific Quality of Service (QoS) to each flow. Since the ad hoc network may frequently change topology, the flow-state mechanisms are directly integrated into the routing protocol to minimize their reaction time and to provide notifications to a flow when the network must break its promise for a specific level of QoS.

2.1.2. QoS routing in mobile ad hoc networks

Due to the fact that network resources are very limited in MANETs, QoS routing is achieved with constraints on bandwidth, delay, jitter, packet loss rate and route stability. The characteristics of MANETs also determine the challenges in ad hoc QoS routing:

- The link capacity is time-varying, which makes admission control difficult.
- Resource reservation is not stable, because the availability of the reserved bandwidth over shared medium is not guaranteed. As mentioned before, the communication capacity between nodes can dramatically change, which may result in QoS re-routing or routing recovery.
- Once a route fails, failure detection and recovery is required.
- End-to-end delay guarantee is not hard in an unsynchronized network.

T. Chen gives a comprehensive description on the problem and current algorithms of QoS routing in ad hoc wireless networks in his Ph.D. dissertation [12]. The defects of the existing routing algorithms are analyzed in the thesis, which include the inability of meeting the requirements of ad hoc wireless networks (such as high accuracy, low overhead, scalability in a large network, the possibility of providing QoS routing etc). The Global State Routing (GSR) approach is proposed. The GSR maintains a global view of network topology and optimizes their routing decisions locally based on the link state vectors exchanged among the neighbor nodes during exchange of routing information. The exchange frequency of link state vectors depends on the node's distance to destination. This multi-level fisheye scope scheme keeps the control message small and therefore reduces the consumption of bandwidth by control overhead.

C. Lin *et al.* proposed a bandwidth routing protocol for QoS support in a multi-hop mobile network [13]. The protocol contains end-to-end bandwidth calculation and allocation. The source is aware of the bandwidth and QoS available to all the destinations in the mobile network. This knowledge enables the establishment of QoS connections within the mobile network and the efficient support of real time applications. The case of ATM interconnection is also discussed in the paper.

A distributed QoS routing scheme is proposed in [14]. In the proposed algorithms, multiple paths are searched in parallel to find the best qualified, which is called "ticket-

based probing". The advantageous properties of the ticket-based probing include dynamic tradeoff between the overhead and the routing performance; working with imprecise state information; avoiding any centralized path computation that could be very expensive for QoS routing in large networks; and the local and end-to-end states maintained at the intermediate nodes can be collectively used to direct the probes along the low-cost feasible paths toward the destination. Fault-tolerance techniques are employed in the scheme for the maintenance of the routing paths resulted from changes of network topology, which enable the proposed algorithms to tolerate high information imprecision. To improve the overall network utilization performance, a heuristic algorithm is proposed for the NP-complete delay-constrained and least-cost routing problem. The algorithm considers QoS constraints as well as cost optimality of the routing path. The simulation results showed that the algorithms achieved a high call-admission ratio, low-cost paths and a modest routing overhead.

Ad hoc QoS On-demand routing (AQOR) [15] is a resource reservation-based routing and signaling algorithm that provides end-to-end QoS support. AQOR includes the following QoS support *features*: (1) accurate measurement of bandwidth availability in the shared wireless channel and accurate measurement of effective end-to-end delay in an unsynchronized environment, (2) distributed routing algorithm that adapts with the dynamic environment, (3) resource reservation that guarantees the available resources, (4) efficient resource release upon route adjustment, (5) instant QoS violation detection and (6) fast and efficient route recovery. AQOR integrates on-demand route discovery between the source and destination; signaling functions for resource reservation and maintenance; and hop-by-hop routing. It introduces a detailed computation of available bandwidth and end-to-end delay. In traffic estimation and bandwidth availability, it considers both self traffic and neighbor traffic to reduce the hidden-node effect, which means that some bandwidth reserved at a certain node is for the traffic between neighboring nodes. AQOR estimates end-to-end downlink delay by measuring round trip delay. AQOR achieves *adaptive routing* by detecting QoS violations at the destination

node or intermediate nodes. Two types of QoS violations are considered in the protocol: (1) channel deterioration in one of the links of the active route, which is an end-to-end delay and detected at destination; (2) route break, which may be cause by left of some node on the active route. This violation can be detected through bandwidth reservation timeout at the destination, or MAC retransmission failure at some intermediate node on the route. The routing adjustment overhead is reduced by employing *destination-initiated recovery*.

CEDAR, a *Core-Extraction Distributed Ad hoc Routing* algorithm for QoS routing in a small to medium size ad hoc network is proposed in [16].CEDAR dynamically establishes "a core of the network, and then incrementally propagates the link state of stable high bandwidth links to the nodes of the core. Route computation is on demand, and is performed by core nodes using only local state." CEDAR has three main components:

- The establishment and maintenance of a self-organizing routing infrastructure called the *core* for performing route computations, which is also called *core-extraction*. A set of nodes are selected distributedly and dynamically to develop the core of the network. In this process, a minimum dominating set of the ad hoc network is estimated using only local computation and local state. Each core node maintains the local topology of the nodes in its domain, and also performs route computation on behalf of these nodes.

- The propagation of the link-state of high-bandwidth and stable links in the core through *increase/decrease waves*, which is also called *Link state propagation*. QoS routing in CEDAR is achieved by propagating the bandwidth availability information of stable links in the core that is known to nodes far away in the network, while information about dynamic links or low bandwidth links is kept local. Slow-moving *increase-waves* and fast-moving *decrease-waves*, which

denote corresponding changes in available bandwidth on links, are used to propagate non-local information over core nodes.

- A QoS route computation algorithm that is executed at the core nodes using only locally available state, which is also called *Route computation*. Route computation first establishes a core-path from the *dominator* (core in the domain) of the source to the dominator of the destination. The core path provides the directional information of the route from the source to the destination. Using this directional information, CEDAR iteratively tries to find a partial route from the source to the domain of the furthest possible node in the core path (which then becomes the source for the next iteration) satisfying the requested bandwidth, using only local information. Effectively, the computed route is a shortest-widest-furthest path using the core path as the guideline.

The *advantages* of CEDAR include the facts that route discovery or maintenance duties are limited to a small number of core nodes, and link state propagation is a function of link stability or quality. The *disadvantages* of CEDAR are: core nodes have to handle additional traffic, which are associated with route discovery and maintenance; and it is hard to converge under high mobility.

2.2. Security in Mobile Ad Hoc Networks

Mobile Ad Hoc Networks are characterized by the absence of fixed infrastructure, rapid topology change and high node mobility. These characteristics determine that wireless ad hoc network is more vulnerable to malicious attacks than the traditional Internet. The vulnerabilities are mainly caused by the following reasons [17]:

- The use of wireless links makes the network susceptible to attacks ranging from passive eavesdropping to active interfering. It’s not like what is in traditional

wired networks that attackers have to physically access the wires or get through several defense lines at firewalls or gateways.

- Mobile nodes able to roam independently makes them easier to be captured, compromised and hijacked. Since tracking down a particular mobile node in a large-scale ad hoc network could be hard, attacks by a compromised node from within the network are far more damaging and much harder to detect. Creating and maintaining trust among peer nodes is also difficult and thus Byzantine failure should be prevented.

- Due to lack of centralized mechanisms in ad hoc network and many algorithms rely on the cooperative participation of all nodes, adversaries can exploit this vulnerability for new types of attacks designed to break the cooperative algorithms.

- Most ad hoc routing algorithms are also cooperative in nature, which is unlike with a wired network, where extra protection can be placed on routers and gateways. Therefore, a compromised node could paralyze the entire wireless network by disseminating false routing information.

Due to these characteristics, the mobile ad hoc networks have harder security requirements than the traditional, wired and static Internet. One of the most severe threats to the routing in ad hoc networks is attack from compromised nodes, which could exert unpredictable and undetectable Byzantine failures .

2.2.1. Routing and network-layer security

MANET routing protocols can be categorized into *proactive*, *reactive* and *hybrid*. *Proactive* schemes try to keep up to date with the topology and routing information in

the network. This achieves low latencies and good routes, since the best path, according to the protocol's metric, should always be known when the node wishes to send a packet. However, it also results in high overhead because the information may change frequently.

Therefore, this can be difficult and expensive for mobile ad hoc networks. Traditional link-state and distance-vector routing protocols are all proactive. MANET proactive routing protocols include DSDV [18], OLSR [19] and etc. *Reactive* schemes only discover routing information as it is needed, or on-demand. This greatly reduces the routing overhead incurred by proactive protocols at the expense of higher latencies, when routes to a requested destination must be discovered before packets can be sent. It can also cause longer routes, since reactive schemes will continue to use an established route as long there are no errors, even if a shorter route appears later due to changes in the topology. MANET proactive routing protocols include AODV [20], DSR [21], TORA [22] and etc. *Hybrid* schemes, such as ZRP [23], CEDAR [16], and etc., use constrained link state maintenance. The routes are also established on demand. Hybrid scheme is proposed in consideration that since ad-hoc network can exhibit quite a range of topology behavior, routing schemes could adapt to a current state of the network, pre-computing routes when mobility is low, and waiting for send requests to initiate route discovery when mobility is high.

A set of design techniques for intrusion-resistant ad hoc routing algorithms are proposed in [24] to protect ad hoc networks against denial of service attacks. These techniques are routing algorithm independent principles that can be incorporated into a number of existing ad hoc routing algorithms to make them robust and resistant to malicious intrusions. The mechanisms aim to limit the damage from intrusion attacks and to allow for continued network operation at an acceptable level during the attacks. The proposed techniques include: *flow-based route access control (FRAC)*, which is used to restrict data traffic passing through a router to authorized flows by means of maintaining an

access control rule base at each router that defines the list of authorized flows allowed to be forwarded by the router; *multi-path routing*, which refers to the ability of ad hoc routing algorithms to discover and maintain all legitimate routes for a data flow; *source-initiated flow routing*, allowing the source to specify which of the multiple paths between the source and the destination will be used; *flow-monitoring*, which enables the detection of path failures resulting from a various types of intrusion attacks (such as data flow disruption attack and resource depletion attack); *fast authentication*, a lightweight mechanism for authenticating data packets flowing through a wireless router that relies on placing the path label of a packet at a node specific secret location within the packet; *sequence numbers*, which counters replay attacks; and *referral-based resource allocation*, to prevent from colluding attacks.

A new routing protocol Ariadne [25] is introduced based on unoptimized DSR, which aims to address routing security. The security mechanism is to protect against *wormhole* attacks, in which two colluding nodes establish "a private, possibly out-of-band, channel" between them and modify routes to go through this link or secretly forward information over it. Adversary model is also proposed based on the number of adversaries and whether they possess cryptographic keys (Byzantine failures) or not. The authors note that most other protocols simply require a single MAC-layer key, which gives no protection against Byzantine failures. Therefore Ariadne requires each pair of nodes to share a unique pre-distributed secret. This secret seeds a PRNG that generates directional confidentiality and integrity keys between each pair. Even though confidentiality keys are set up, they are not explicitly used in the Ariadne protocol. The protocol relies on the integrity/authentication keys and the TESLA authentication scheme, which is proposed by Perrig et al. [25]. Messages are sent with authentication codes under the TESLA keys, which are generated from a reversed one-way hash chain. Each key is valid for a certain period of time and is disclosed only after the time period is finished and the key is invalid.

Authenticated Routing for Ad hoc Networks [26] (ARAN) is more an authentication scheme than a routing protocol. It depends on public key certificates and trusted common certificate authorities to provide authentication for routing process. It roughly defines a path discovery method, but does not specify how routing information is kept in the packets not at the nodes. ARAN defines two levels of authentication, an end-to-end authentication service which includes hop-by-hop authentication for only the current hop; and an all-to-end authentication service (Shortest Path Confirmation), in which all hop-by-hop authentications are preserved and the packets are also re-encrypted under the destination's public key. In both cases the relevant certificates are included in the packet; two certificates for the end-to-end case, *n* for the all-to-end case, where *n* is the number of nodes the packet has visited. Due to the fact that certificates are often large, both methods are quite expensive for energy constrained devices, as are often found in ad-hoc networks.

To combat the problem of the heavyweight cryptographic requirements in ARAN protocol, LARAN (Lightweighted Authenticated Routing for Ad hoc Networks) is proposed . LARAN uses lightweight cryptography via one-way hash chains to achieve “nearly double the performance of ARAN with only minor impact upon security considerations”. Due to the use of one-way hash chains for authentication, the LARAN protocol requires that “a packet sent by a node is received by a neighboring node before a third node can replay the packet to it, unless the neighbor under consideration has dropped the packet”. Analysis is also provided to address the security solutions against attacks that introduced when attempting to move to lightweight hash-chain based security. These attacks are: Bootstrap replay attack, FAIL message flooding attack, node movement attack, tunneling bootstrap attack, and jamming attack.

Hop-by-hop authentication is a widely used security mechanism in the traditional Internet for protecting such features as *integrity*, *confidentiality* and *nonrepudiation*. The Secure Ad hoc On-Demand Distance Vector Routing (SAODV) [27] is an extension of

the AODV [20] routing protocol that exploits hop-by-hop authentication to protect the route discovery mechanism providing these security characteristics. SAODV requires all intermediate nodes cryptographically validate the digital signature appended with the routing messages and consequently imposes a high overhead on routing process.

Hop-by-hop authentication is neither efficient nor effective due to its extensive overhead as well as the fact that authentication can only identify a node but can not determine whether the information distributed by the node is correct. A malicious node inside the network could raise false alarms or send false link state information, which would result in other nodes being wronged or the network paralyzed. Deliberate distribution of false information could be far more damaging and much harder to detect than other forms of attacks.

To battle the high process overhead in SAODV, the Secure Routing Protocol (SRP) for ad hoc networks was proposed by Papadimitratos and Haas [28]. SRP assumes only the source and destination nodes are trusted and thus securely associated, which removes the overhead on intermediate nodes. The protocol guarantees that fabricated, compromised, or replayed route replies would either be rejected or never reach back to the querying node. SRP achieves robustness in the presence of noncolluding nodes, and provides accurate routing information in a timely manner.

SEAD (Secure Efficient Ad Hoc Distant vector routing protocol) is a secure ad hoc routing protocol based on the design of the Destination-Sequenced Distance-Vector routing protocol (DSDV) [29]. One-way hash functions instead of asymmetric cryptographic operations are used in the protocol in order to support use with nodes of limited CPU processing capability, and to guard against Denial-of-Service (DoS) attacks in which an attacker attempts to cause other nodes to consume excess network bandwidth or processing time. The simulation results showed that SEAD is robust with

presence of multiple noncolluding attackers creating incorrect routing state in any other node.

While Quality of Service (QoS) is being regarded as another critical service other than security in ad hoc networks, researchers are trying to take advantage of both mechanisms by embedding one into the other. [30] presents a Security-Aware Ad Hoc Routing (SAR) protocol that incorporates security attributes as parameters into ad hoc route discovery. SAR employs the idea of "quality of security" and ensures data are routed through a secure path only composed of nodes at the same trust level. The authors suggested to simply mirror organizational privileges in trust level establishment and to encrypt the portion of the RREQ and RREP headers that contain the trust level. The desirable security properties associated with the "level of protection" include *timeliness*, *ordering*, *authenticity*, *authorization*, *integrity*, *confidentiality* and *nonrepudiation*. While routing through a set of trusted nodes guarantees greater security, it is not always feasible to find a path that only includes nodes at the desired trust level.

With these remarkable probes, it has been proved extremely difficult to find a panacea achieving both effectiveness and efficiency for ad hoc routing security. All approaches have to make a tradeoff between these two performances.

A network-layer security solution in ad hoc networks is described in [31], which protects both routing and data packet forwarding functionalities in the context of the AODV protocol. The proposed self-organization approach employs a "full-localized" design, which does not assume any priori trust or secret association between nodes. In the design, each node has a token to be temporarily admitted to the network, which will expire and has to be renewed. The period of the validity of a node's token is dependent on how long it has stayed and behaved well in the network. The behavior of the node is monitored collaboratively by its local neighbors, and any misbehavior in routing or packet forwarding services will be detected. To improve the monitoring accuracy and

withstand the blackmail attack, ‘m out of N’ strategy is used to cross-validate the monitoring results of different nodes in the neighborhood.

The advantages of this “full-localized” design include: the local information is more credible than that from the remote hosts because, therefore the detection accuracy of the security mechanisms that use local information should be higher than those using remote information; Secondly, “full-localized” approach removes the necessities of propagating security information between detecting nodes with multi-hop distance and therefore reduces network traffic. The disadvantages of the design are: the detection is only effective within the neighborhood and therefore does not perform well in presence of mobile attackers; the audit data is limited to local information; and overhearing the channels could be unreliable in some circumstances and consequently the detection is prone to attacks on data-link channels.

A new mechanism, called *packet leashes*, is presented in [32] for detecting and defending against wormhole attacks. Wormhole is a severe attack in ad hoc networks that is possible even if the attacker has not compromised any nodes, and even if all communication provides authenticity and confidentiality. In the wormhole attack, an attacker records packets at one location in the network, and then tunnels them to another location, and retransmits them there into the network. Because the wormhole attack is particularly dangerous against many ad hoc network routing protocols in which the nodes that hear a packet transmission directly from some node consider themselves to be in range of (and thus a neighbor of) that node, it is important for a node to know how far it is away from the sending node. Two kinds of leashes are proposed:

- *Geographical leash*, in which each node must know its own location and all nodes must have loosely synchronized clocks. When sending a packet, the sending node includes in the packet its own location, and the time at which it sent the packet; when receiving a packet, the receiving node compares these values to

its own location, and the time at which it received the packet. The receiver can compute an upper bound on the distance between the sender and itself.

- *Temporal leash*, in which all nodes must have tightly synchronized clocks, such that maximum difference between any two nodes' clocks is c. The value of the parameter c must be known by all nodes in the network, To use temporal leashes, when sending a packet, the sending node includes in the packet the time at which it sent the packet; when receiving a packet, the receiving node compares this value to the time at which it received the packet. The receiver is then able to detect if the packet traveled too far, based on the claimed transmission time and the speed of light.

A specific protocol, called TIK, which implements leashes is also presented in the work.

AODV protocol can be vulnerable to impersonation attacks [33]. A malicious node can issue in-the-middle attack, to hijack the traffic from node A and then communicates with A while pretending to be node B, which is the real destination of node A's traffic. The authors point out that the classic approaches where public key cryptography can not be applied in mobile ad hoc networks due to the fact that a central authority is not available. Instead, this approach uses Cryptographically Generated Identifiers and Addresses that are derived from the hash of the node's public key, which are "statistically unique" and "securely bound to a given node". Therefore, it allows two hosts A and B to establish a secure channel over an insecure ad hoc network that uses AODV.

2.2.2. Intrusion detection architectures

Generally, an Intrusion Detection System can be classified based on the detection technique as described below:

- *Signature-based* (or misuse) detection monitors for the occurrence of predefined signatures or sequences that indicate an intrusion. The advantages of this technique are that they have the potential for very low false positive rates, and the contextual analysis is detailed, which makes it easier for the people who are using this detection system to take preventive or corrective action. But the drawback of this approach is that it does not perform well at detecting previously unknown attacks.

- *Anomaly-based* detection defines a profile of normal or expected behavior and classifies any deviation of that profile as an intrusion. The normal profile is updated as the system learns the subject's behavior. This technique may detect previously unknown attacks, but may exhibit high rates of false positives.

- *Specification-based* detection defines a set of constraints that describe the correct operation of a program or protocol, and monitors the execution of the program with respect to the defined constraints. This technique may provide the capability to detect previously unknown attacks, while exhibiting a low false positive rate.

IDS solutions for fixed wired networks are often hierarchical and deploy network-based sensors at key traffic concentration points, such as switches, routers, and firewalls. These IDS sensors are physically secured, and use the signature-based detection technique to detect attacks. Alerts generated by these distributed IDS sensors are sent to centralized security servers for analysis and correlation. The effectiveness of IDS solutions that were designed for fixed wired networks are limited for wireless ad-hoc networks due to the following reasons [34]:

- Absence of key concentration points in wireless ad-hoc networks where network traffic can be monitored limits the effectiveness of a network-based IDS sensor,

because only the traffic generated within radio transmission range may be monitored.

- It may be difficult to depend on the existence of a centralized server to perform analysis and correlation in a dynamically changing ad hoc network.

- The secure distribution of signatures may be difficult, due to the properties of wireless communication and mobile nodes that operate in disconnect mode.

Y. Zhang *et al.* proposed an architecture for intrusion detection in mobile wireless networks and evaluated anomaly detection through simulation experiments [17]. In this architecture, if a node detects an intrusion with weak or inconclusive evidence, it can initiate a cooperative global intrusion detection procedure, or if a node detects locally an intrusion with strong evidence, it can independently determine an attack on the network. In the intrusion detection architecture, a conceptual model for an IDS agent is described that is composed of six units: *local data collection*, *local detection engine*, *cooperative detection engine*, *local response*, *global response* and *secure communication*. The anomaly detection model employs the following procedure: 1) select audit data to obtain a low entropy for the normal dataset; 2) perform appropriate data transformation; 3) compute classifier using training data; 4) apply the classifier to test data; and 5) post-process alarms to produce intrusion reports. Two classifiers, RIPPER and SVM Light, are studied via simulation. The simulation results showed that RIPPER performed poorly, which indicated that “quasi-linear anomaly detection analysis used in traditional intrusion detection systems can not be used in ad hoc networks”, because high mobility defeated such effort.

The shortcoming of a cooperative and distributive IDS architecture is that it could be susceptible to attacks from Byzantine nodes, which could independently make false

claims of detecting an attack from a correct node with strong evidence, thus making it difficult to derive a distributed consensus.

Hierarchical IDS architectures have been proposed for multi-layered, wireless ad-hoc networks. In a multi-layered wireless ad hoc network, cluster-head nodes centralized routing for the cluster and may support additional security mechanisms. [35] is a three layered infrastructure that may be deployed in the tactical battlefield, consisting of two-layered ground networks and a third layer of Unmanned Aerial Vehicles (UAVs). The UAVs provide event correlation for a theater of operations. Neighboring ground nodes detecting that ground node V is acting malicious send an accusation message to the UAV, the UAV will determine that node V is compromised after receiving a threshold of K accusations. Then the UAV may respond, such as broadcasting a message to notify all nodes in the theater. In this way, a UAV acts as a central security decision point for the network.

2.2.3. Key management

Key exchange and distribution is also a significant issue in MANET security. Key management's goal is to establish a shared secret between all participating parties. There are several methods of achieving this, namely key predistribution, key transport, which includes arbitrated keying schemes, and key agreement. Each of these has benefits and problems in the ad hoc wireless setting. Key predistribution requires the least communication and computation to establish a common key; a node either has a key, or it doesn't. Arbitrated keying requires less prior configuration but more messages and computation. These protocols often require network synchrony and have a single point of failure in the arbitrator, which is not very practical for wireless ad hoc networks. To circumvent this, the service may be distributed to several nodes (e.g. in [36]), in which more pre-configuration is required and some of the scheme's benefits are lost.

The absence of a fixed topology and a central authority makes it difficult to implement key management in mobile ad hoc networks. Some solutions have been provided in [37, 38], in which approaches are proposed such as key generation, issue, storage and distribution of public-key certificates.

2.2.4. Unique identification

Since the topology of a mobile ad hoc network is dynamic and self-organizing, a node can join and leave the network at any time. Therefore, the maintenance of the identifiers (or addresses) of the nodes becomes a problem. If a node does not have a unique and recognizable identifier, it can escape from the punishment even if it is detected as misbehaving. The following work may help to solve this problem.

It has been noticed that in both of the two approaches that can be used to ensure the uniqueness of an address (the Duplicate Address Detection (DAD), usually done by sending a query to the chosen address and waiting for a response; and the distributed assignment of a priori unique addresses, which can also be a bandwidth consuming task in a dynamic environment) a merger of two configured networks is very difficult to detect and can lead to duplicate addresses. Thus, a continuous and bandwidth-efficient duplicate address detection mechanism would be eligible. In this paper, a new DAD approach is proposed [39]. In the new approach, the detection of duplicate addresses in a passive way, only by monitoring routing protocol traffic.

A “unilateral authentication protocol” is proposed in [40] to protect IPv6 networks against abuse of mobile IPv6 primitives. A mobile node uses a partial hash of its public key for its IPv6 address. This protocol integrates distribution of public keys and protects against falsification of network addresses. Although it is targeted at mobile networks with stations, it can also be used in ad hoc networks.

Some researchers solved the identifier problem using characteristics of Statistic Uniqueness and Cryptographic Verifiability (SUCV) of certain entities, which characteristics allow them to severely limit certain classes of denial of service attacks and hijacking attacks [41]. The idea is to use identifiers that have a strong cryptographic binding with their public components (of their private-public keys).

2.3. QoS Security in Wired Networks

Without the protection from security mechanism, QoS can be vulnerable to various attacks. An attacker's objectives can be one or more of the following:

- Denial of QoS request. This can be achieved by intentionally dropping or delaying reservation messages; spoofing *teardown* message can also result in QoS reserved be cancelled by illegal host.

- Degradation of network utilization. An attacker can maliciously alter QoS signaling packets, which may result in unnecessary or suboptimal resource reservations

- Reserved QoS degradation. Even if QoS resources have been reserved along the path, a malicious node on the path can still use the reserved resource without proper authorization; or drop or delay data packets intentionally, which may result in degradation on reserved QoS. Although QoS violation detection mechanisms have been used in a few QoS approaches, the affect may not be recovered in a short period of time.

Integrated Services (IntServ) and Differentiated Services (DiffServ) are two models proposed to support QoS in networking. RSVP is a signaling protocol for resource reservation in IntServ model, which allows some users to access preferential networking

resources. RSVP security issues include: node and user authentication, message integrity, confidentiality, non-repudiation, replay attacks and DoS attacks [42].

RSVP cryptography authentication mechanism is used to protect RSVP message integrity hop-by-hop [6, 43]. An INTEGRITY object is defined to be carried in RSVP message in order to provide the information required for hop-by-hop integrity checking, which helps to protect RSVP messages against spoofing and corruption. Hop-by-hop authentication cannot prevent attacks by the RSVP nodes on the path, named as *insider.*

Tsung-li Wu *et al.* proposed a secure RSVP protocol, Selective Digital Signature with Conflict Detection (SDS/CD) for RSVP, which combines attack prevention and intrusion detection [44, 45]. The protocol can deal with insider attacks that cannot be countered by the RSVP authentication cryptography. They described attacker's objectives as *Denial of QoS service request*, *unnecessary/suboptimal resources reservation*, *degradation of network utilization*, and *reserved QoS degradation*. The algorithm can be simplified as follows:

- $Sender_{Alice}$ selectively and digitally signs *Tspec(PATH)* with her private key, $Receiver_{Bob}$ verifies with *Alice*'s public key;

- $Receiver_{Bob}$ sends *RESV* piggybacking with historical *Adspec(PATH),* digitally signed with *Bob*'s private key;

- Intermediate RSVP-enabled router $Router_{Chris}$ verifies if piggypacked *Adspec(PATH)* is less than or equal to the forwarded *Adspec(PATH);*

- *SenderAlice* uses Bob's public key to verify if *Rspec(RESV)* is correctly signed by *Bob;*

- Similar procedure for refresh messages;

- Once a node (or router) detects something abnormal, it sends alarm to the Policy Decision Point, which will decide whether to issue intrusion response to the misbehaving node.

Vanish Talwar *et al.* proposed an RSVP-SQOS (RSVP with Scalable QoS protection) protocol [42, 46], which targets at the drawbacks of high overhead and bad scalability respectively in [6] and [44]. To make the design scalable, RSVP-SQOS divides the network into domains or sub-networks and modifies the algorithm in [44]. RSVP-SQOS adopts the same idea of digitally signing the non-mutable messages and checking the integrity of mutable parameters via feedback messages. The intermediate ingress routers as well as the receivers can also generate feedback messages to verify the integrity, which will be used to detect malicious attacks during inter sub-networks delivery.

Some researchers have been seeking to prevent against both types of vulnerabilities and attacks on QoS: attack on *control flow* and attack on *data flow* [47]. The work is composed of two parts: resource pricing, which protects control flow, and analysis of TCP dropping against attacks on data flow. In resource pricing, the problems that are dealt with include unauthorized use of resources and denial of access to an authorized user. The *demand-based pricing* method is used as the solution. The article states that packet dropping attack is one of the most difficult attacks to handle among the various types of denial of service (DoS) attacks. There are three packet-dropping patterns: *periodic dropping*, *retransmission-based dropping* and *random dropping*. The detection is conducted at end systems instead of requiring cooperation from other nodes in the network.

QoS routing is to find a suitable path through the network between the sources and destinations that will have the necessary resources available to meet the QoS constraints

for the desired service, and to set up the resource reservation along the path. A comprehensive reference for QoS routing problems can be found in [48]. QoS routing, dependent on the accurate availability of the current network state, could exact security problems because compromised nodes could provide false information or provide expired information via replaying old routing messages. Distributed or hop-by-hop routing can also introduce new security problems since the source and other nodes are involved in path computation by identifying the adjacent router to which the source must forward the packet associated with the flow.

2.4. QoS Security in Mobile Networks

There are publications [49] [50] aimed at providing different levels of security among different groups of nodes while integrating security into Quality of Service as a parameter.

Cluster based Routing for End-to-end Security and Quality of service satisfaction (CRESQ), a new QoS routing protocol for ad hoc networks, is proposed in [49]. It uses clustering to minimize the routing overhead and uses localized route recovery to minimize route and QoS re-establishment delay. In CEWSQ, a route is established with the involvement of intermediate clusters, which means the routing algorithm interactions take place at the cluster level. It considers QoS parameters before making a connection. The source node is aware of the intermediate nodes and the information will be used in case security is desired. The source node may specify levels of security (such as authentication, encryption and etc.) in the QoS specification.

As we have mentioned, SAR protocol is presented in [30] and [50] . The protocol also incorporates security attributes as parameters into ad hoc route discovery. SAR employs the idea of “quality of security” and ensures data are routed through a secure path only composed of nodes at the same trust level.

3. A LIGHTWEIGHT AUTHENTICATION PROTOCOL FOR MOBILE AD HOC NETWORKS

Most ad hoc networks do not employ any network access control, leaving them vulnerable to resource consumption attacks. In ad hoc networks, users need to assure the party who supposedly sent a message to another party is indeed the legitimate party. Otherwise, a malicious node could tamper a network with falsified data. These attacks can result in degraded performance of networks, interference of resource reservation, and unauthorized use of resources.

Authentication mechanisms are used to ensure that the entity who supposedly sent a message to another party is indeed the legitimate entity. General security requirements for authentication include protection against replay attacks, resistance against man-in-the-middle attacks and provision of confidentiality. There are two basic kinds of cryptography that have been widely used for the traditional Internet: *symmetric* cryptography and *asymmetric* cryptography (such as digital signature).

Different from the fixed networks, the communication links in mobile ad hoc networks are open shared medium, which makes the communications between neighboring nodes more vulnerable to attacks such as packet forging and malicious alteration. In addition, mobile ad hoc networks are characterized by absence of fixed infrastructure, rapid topology change and constrained resources (such as limited battery power, small computational capacity and bandwidth). These characteristics determine that the authentication protocols used for routing and data packet delivery in MANETs should be lightweight and scalable. Asymmetric cryptography does not adapt well to MANETs in that the processing required for asymmetric cryptography is very CPU intensive and the technique has been proved to be prohibitively insufficient in wireless ad hoc networks in terms of message overhead and computation complexity. Symmetric cryptography

algorithms are fast. Nevertheless, it introduces complexity in key maintenance and exerts difficulty in authentication for multicast or broadcast communications.

Moreover, radio channels in wireless networks are more erroneous and lossy than the communication links in the Internet. With multiple receivers, there could be a high variance among the bandwidth and radio interference of different receivers, with high packet loss for the receivers with low bandwidth and high radio interference. In consideration of this problem, the authentication mechanism is expected to be effective even in the presence of high packet loss.

The idea of TESLA key is proposed in [51].TESLA uses one-way hashed chain to generate keys, and delays disclosure of keys to guarantee that a node receives the packet before another node can forge the packet with already released keys. But the security condition of TESLA requires clock synchronization, which is very difficult to achieve in mobile ad hoc networks, if not impossible.

The design of our protocol is motivated by LHAP (a Lightweight Hop-by-hop Authentication Protocol for Ad Hoc Networks) [52]. LHAP is a lightweight hop-by-hop authentication specially designed for ad hoc networks. It uses two keys: TRAFFIC key and TESLA key. TRAFFIC key is used to authenticate packets; and TESLA key is used to achieve trust maintenance by authenticating KEYUPDATE message. KEYUPDATE message is sent periodically to guarantee that the current released key is valid so that a malicious node will not be able to use an obsolete key to forge a packet. LHAP is not only a comprehensive authentication approach, by thoroughly describing key management and traffic authentication, but also proved to be computationally efficient. However, it requires two keys, which hence not only adds more complexity in authentication, but also needs to periodically send key maintenance packages that themselves need to be authenticated with TESLA keys. In addition, LHAP does not eliminate the disadvantage of delayed authentication in TESLA because the authenticity

of the packets and the TRAFFIC key can not be verified until TESLA key is authenticated.

In this chapter, we will propose a lightweight authentication protocol, which utilizes *one-way hash chain* to provide effective and efficient authentication for neighboring communications in MANETs. Our protocol is lightweight, scalable and tolerant of packet loss.

3.1. The Authentication Protocol

This authentication protocol utilizes *one-way hash chains*, which is more efficient and less expensive than asymmetric cryptographic operations. One-way hash chain is a widely-used cryptographic primitive that uses a *one-way hash function* to generate a sequence of random values that serve as authentication keys. It has been used in authentication schemes for wireless ad hoc networks [29] and sensor networks [53].

Figure 1 demonstrates the one-way hash chain construction, utilization and revelation. To generate a key chain of length n+1, the first element of the chain h_0 is randomly picked and then the chain is generated by repeatedly applying a one-way function (denoted as H in Figure 1). A one-way hash function maps an input of any length to a fixed-length bit string, which is defined as $H:\{0,1\}^* \rightarrow \{0,1\}^{\phi}$, where ϕ is the length of the output of the hash function – the newly generated key. The function H should be simple to compute, nonetheless must be computationally infeasible in general to invert. In utilization and revelation of these keys, we use the reverse direction of key generation: we start from h_n, the last generated, and then h_{n-1}, …, h_0. Any key of the one-way key chain commits to all previous keys[2], and h_n is a commitment to the entire one-

[2] In the dissertation, when we refer to the direction of key generation as the direction of the chain. For example, the subsequent key of h_0 is h_1, and so on.

way chain. Any key h_j can be verified from h_i ($0 \le i < j \le n$) to be indeed an element in the chain by repeatedly applying H for $j - i$ times, that is, $h_j = H^{j-i}(h_i)$. Therefore, given an existing authenticated element of a one-way hash chain, it is possible to verify elements later in the sequence of use within the chain.

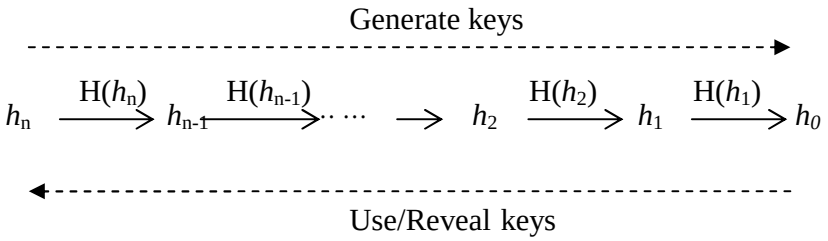


Figure 1. An example of one-way hash chain

The chain of keys can be created all at once off-line before the mobile node joins the network and then stored for later use.

We use the following notations to describe our authentication protocol (see Table 1).

Table 1. Notation for Authentication Protocol

Symbol	Description
A, *B*	Identities of mobile nodes
Cert_A	Certificate of node A's public key signed by CA's private key
$Sign_A(M)$	Digital signature of message M, signed with node A's private key
MAC(M, *K*)	MAC over message M with key *K*
h_i^A	The i[th] key in node A's one-way hash chain
H_A	Node A's hash function
H_A^k	Applying A's hash function for *k* times
M1|M2	The concatenation of message M1 and M2
P_i^A	The i[th] packet of node A's.

In this section, we will discuss the assumptions on which our protocol is established, which is followed by a detailed description on the basic scheme of our authentication protocol, including trust management and message authentication; and at last we will address the problem of key disclosure.

3.1.1. Assumptions

To prevent a malicious entity from forging packets with MACs that are computed using already released key, a packet sent by a node has to be received by an immediate neighboring node before a third party is able to replay the packet to it, unless the receiver has dropped the packet. This necessary condition for authentication using one hash key chain is assured in our approach by using delayed key disclosure. The scheme of key disclosure will be discussed later in this chapter.

We assume that each node can communicate with a trusted certificate authority (CA) before it enters the ad hoc network, and it can obtain a public key certificate signed by the CA as well as an authentic public key of the CA. The public key of the CA will be used to verify key certificates distributed by other nodes. However, a node may not be able to contact the CA after it joins the network because it is difficult for an ad hoc network to provide a central administration point since all the nodes in an ad hoc network are mobile. Moreover, a central entity is very likely to become the most vulnerable point in the network, which is subject to various malicious attacks.

We also assume that the mobile nodes that we are protecting are relatively underpowered so that asymmetric key operations such as digital signatures are too expensive for them to compute for each packet. In our scheme, digital signature is only used in trust bootstrapping so that the nodes can verify the genuineness of the first revealed key. Once the initial key is confirmed to be authentic, the subsequent keys can be verified by applying the one-way hash function.

On the contrary, the adversaries are powerful with the following capabilities: (1) an adversary can be capable of various attacks: eavesdrop, delay, drop, replay or alter packets; (2) an adversary's computation resources can be very large but yet limited. This means that an adversary may be able to conduct fast computations, such as computing MACs with negligible delay. The adversary, nevertheless, cannot invert a hash function and hence cannot obtain a hash key before the key owner reveals it.

3.1.2. Trust management

1) Trust bootstrapping

To use one-way hash key chain for authentication, a node needs to distribute an authentic key such as h_n, which is the first revealed key from its generated chain. This key commits to the whole key chain and therefore the genuineness of the subsequent keys can be verified by applying hash function to this key, such as: given a key h_i, it is a genuine key from the chain if $h_n = H^{n-i}(h_i)$; if not, it is a counterfeit key.

Our scheme requires that a node contact the certificate authority to obtain public key of the CA as well as the certificate of the node's own public key before it joins an ad hoc network. The node can also pre-compute the whole one-way hash key chain off-line to reduce computational latency. Then the node signs the message with its private key and broadcasts a JOIN message to its neighbors. We suppose that a node, say node *A*, is sending JOIN message to its neighbors. The JOIN message will be in the following format:

$$A \to *: Cert_A, \{A \| h_n^A \| H_A\}, Sign_A(A, h_n^A, H_A)$$

where CertA denotes the certificate of node A's public key that has been signed by CA's private key; *A* denotes the identity of node A; and $Sign_A(A, h_n^A, H_A)$ denotes the digital signature of message $\{A \| h_n^A \| H_A\}$.

Upon receiving this JOIN message, every receiving node first uses CA's public key to verify the certificate of A's public key. Once the genuineness of node A's public key is confirmed, the key can be used to verify the digital signature on A's message. If the digital signature is validated to be authentic, the receiving node will record A's initial key h_n^A as well as its hash function H_A.

To bootstrap an authentic hash key to node A, each of its neighbors (say node *B*) unicasts the following ACK message to node A:

$$B \rightarrow A : Cert_B, \{B \| h_m^B \| H_B\}, Sign_B(B, h_m^B, H_B)$$

where h_m^B denotes B's most recently released key. Node A will perform the same verifications on B's ACK message as what node B did with A's JOIN message.

2) Trust maintenance

The trust relationship between a node and its neighbors is maintained with a periodical broadcast of KEYUPDATE message. In the KEYUPDATE message, a key that has been used to compute MACs will be released, and the neighboring nodes will verify the new released key with corresponding hash function. The maintenance process is described below:

Each node periodically broadcasts a KEYUPDATE message to its neighbors, which discloses its most recently used key:

$$A \rightarrow * : A, h_j^A$$

The key h_j^A will be authenticated by its neighbors based on the previously released key h_{j+1}^A: if it can be proved that $H_A(h_j^A) = h_{j+1}^A$, the key h_j^A is considered valid;

otherwise, the key is invalid and the receiving node may optionally issue an intrusion alert to other nodes.

3) Trust termination

In our authentication scheme, the trust relationship between two nodes may be terminated under two circumstances. First, when a node is detected to be compromised, the detecting nodes will permanently terminate their trust relationship with the compromised node. In this case, a further step such as excluding the node from the network might be taken. Second, when a node does not receive the KEYUPDATE message from a neighbor for a period that exceeds a predefined threshold, it will terminate its trust of the neighbor temporarily. This can happen when the neighboring node moves out of the node's transmission range, or when the neighboring node is not transmitting any data packets for a fairly long time (we assume that in case a node does not have any packets to send, it will not release key periodically in order to save its keys). If the two nodes want to restart their communications, they can run the trust bootstrapping process again to reestablish their trust relationship. The value of the threshold is dependent on the size of the cache for authentication at the node. The cache is used to store the authentication information of other nodes', such as hash function, previously released key, and non-verified messages. A node with a larger cache can store more commitment information and therefore a trust relationship may be kept for longer time.

3.1.3. Message authentication

When a node wants to send a message, it computes the MAC on the message and then unicast to the receiving node (say node B), or multicast (or broadcast) the packet (denoted as P^A) to the receivers in the following format:

$$A \rightarrow B(*) : M,\ MAC(M, h_i^A)$$

where h_i^A is the currently used key of node A's. Note that the key h_i^A has not been disclosed at this point. The originator of the packet (node A in this case) will later disclose h_i^A in KEYUPDATE message. The key enables the receiver to verify the MAC of the message. If the verification is successful, the message is then authenticated and trusted. Once the key is disclosed, it becomes obsolete and can not be used to generate MACs any more.

3.1.4. Key disclosure

1) Security condition and threat model on authentication

This authentication protocol can be compromised if an adversary obtains node A's secret key h_i^A before a receiver receives the data packet that is protected with this key, because the adversary would be able to change the message and then use the key to recompute the MAC of P^A, or even to forge all subsequent traffic. To prevent from this type of attacks, the receiver needs to be assured that it receives the data packet before the corresponding key is disclosed by the sender. The following *security condition* describes this requirement:

"A data packet *P* arrived safely, if the receiver receives the packet when the sender did not yet send out the corresponding key disclosure packet."

It is known that radio channels in MANETs are more prone to error than those in the Internet because wireless communication links use open shared medium. The erroneous communication caused by signal conflicts may result in deteriorations of packets or even packet drops.

Figure 2 exemplifies an attack that takes advantage of KEYUPDATE packet drop to send maliciously modified or forged packets. Suppose node A is sending a message *Ms* to its neighbors with MAC (denoted by MAC(Ms, *K*) in Figure 2), which was generated with key *K*. Then A discloses key *K* to its neighbors B, C, D and M. Suppose node B does not immediately receive the message *Ms* and the KEYUPDATE message due to signal conflict at its channel. Node M, which is a malicious entity, then takes advantage of this chance to modify the message to *Ms*' and sends the tampered packet to node B with a MAC that is generated using the disclosed key *K* (denoted by MAC(Ms', *K*) in Figure 2). Node B would believe that it is a legitimate packet from A when it later receives the resent KEYUPDATE message from A (or a replayed KEYUPDATE message from node M).

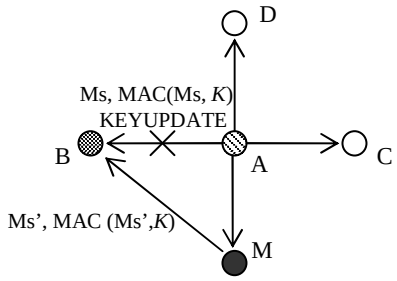


Figure 2. An example of in-the-middle attack on key disclosure

2) Delayed key disclosure

To prevent from the "in-the-middle" attacks described above, a receiver should have the knowledge of when to expect a KEYUPDATE message. TESLA uses delayed key disclosure to solve the problem. It also uses *time synchronization* to guarantee that the receiver can unambiguously verify if the security condition holds on each packet and then decide to keep or drop the packet. However, clock synchronization relies on two assumptions: first, the nodes to be synchronized have the ability to periodically exchange

messages; and second, the nodes have the ability to estimate the time it takes for a message to travel between them. In mobile ad hoc networks, the high mobility of nodes lead to frequent reconfiguration of topology and frequent change of communication capacity between two nodes. Therefore, clock synchronization is very difficult (if not impossible) to achieve in a MANET in that there is no central control and packet delays may vary due to unpredictable mobility and radio interference.

Our authentication protocol uses delayed key disclosure without requirement for clock synchronization. In the protocol, a currently used key is broadcast after the key has been used to generate or verify MACs for a *time interval*. This time interval, namely *delay of key disclosure* in this context, is determined by the sender and announced in the data packets that are protected with the key. Before a key is disclosed, the packets with MACs that are computed with the key cannot be authenticated. Packets can be stored in cache at the receiving node until the key has been received and the authentication is completed.

We define the *delay of key disclosure*, denoted by *d*, as the time difference between key disclosure and the time when sender *starts* to send messages that use the key to compute MACs. Specifically, if a sender starts to send the first packet that is authenticated via MACs with key *K* at time t_0, then key *K* will be disclosed at time $t = t_0 + d$. Suppose there are *m* packets on which MACs are computed with key *K*: denoted by $P_1^K, P_2^K, \ldots, P_m^K$ respectively in sequence of being sent, and the times when they will be sent are $t_1^K, t_2^K, \ldots, t_m^K$ respectively. We denote the time interval between sending of the packet and the key disclosure as *r*, and the interval for packet *i* as r_i. The timeline is shown in Figure 3.

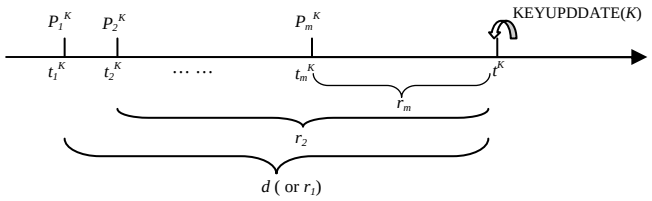


Figure 3. An example of the timeline for a delayed key disclosure

In the example demonstrated in Figure 3, we have:

$$r_1 = t^K - t_1^K (= d);$$
$$r_2 = t^K - t_2^K;$$
$$\cdots\cdots$$
$$r_m = t^K - t_m^K.$$

In our protocol, a sender announces the remaining time r in its data packets[3]. The receiver can estimate when to expect the arrival of the KEYUPDATE message according to the remaining time. Suppose the receiver receives the packet $P_i^K (1 \le i \le m)$ at time rt_i^K. The remaining time indicated in the packet is r_i. In case that the data packet and the KEYUPDATE message are delivered at the same transmission rate, the KEYUPDATE message should arrive at the receiving end at time $rt^K = rt_i^K + r_i$. If the data packet and the KEYUPDATE message are delivered at different transmission rate (supposedly the difference is δ), then the KEYUPDATE message should be expected at the receiving end at $rt^K = rt_i^K + r_i + \delta$. δ can be estimated at each node according to its observation on the traffic .

This scheme eliminates the need for clock synchronization, which is used in TESLA. Although it still needs to estimate the difference between transmission rates of a data

[3] Note that the time when a packet will be sent can not be exactly known at the time of packet generation. However, it can still be accurately predicted according to the cache status at each node.

packet and its KEYUPDATE message, it is easier than clock synchronization because it does not need to estimate the absolute value of transmission delay. Instead, it only needs to estimate the variance of the transmission delays on data packets and the corresponding KEYUPDATE message, which is much easier.

In our protocol, it is possible that a key (say h_i) is disclosed after the packets using the next key h_{i-1} have been sent. Therefore, the receiver needs to know which key is used for which packets. To solve this problem, we include the index of the key in data packets, so that the receiver will be able to know which key should be used to authenticate the message. Therefore, a data packet from node A destined to all its neighbors (broadcast) or to node B (unicast) is in the following format:

$$A \rightarrow *(B) : M, MAC(M), r, index$$

where *index* denotes the index of the key that will be used to authenticate the message. And the KEYUPDATE message will be:

$$A \rightarrow * : A, \quad h_j^A, index$$

The index of the key is not protected in the message. In case that it is tampered such as maliciously increased or decreased, it can still be verified by repeatedly applying hash functions to the key until the result matches the previously received key and meanwhile counting how many times the function has been applied. For example, if the newly arrived key is K and the previously received key is K' and $K' = H^n(K)$, then $index(K) = index(K') + n.$

Using this method, our protocol is tolerant of packet loss because the key verification is not based on the immediate previous key.

In our scheme, the delay of key disclosure can vary for different keys. It is not a predetermined and unchanging value since establishment of the trust relationship, as what TESLA has used. The advantage of varying delays of key disclosure is that it allows a sender to choose key disclosure period according to the pattern of the traffic transmitted by the sender: when the traffic is heavier, the delay should be smaller; and vice versa. This can prevent the cache from being "flooded". An example of this varied delays scheme is demonstrated in Figure 4.

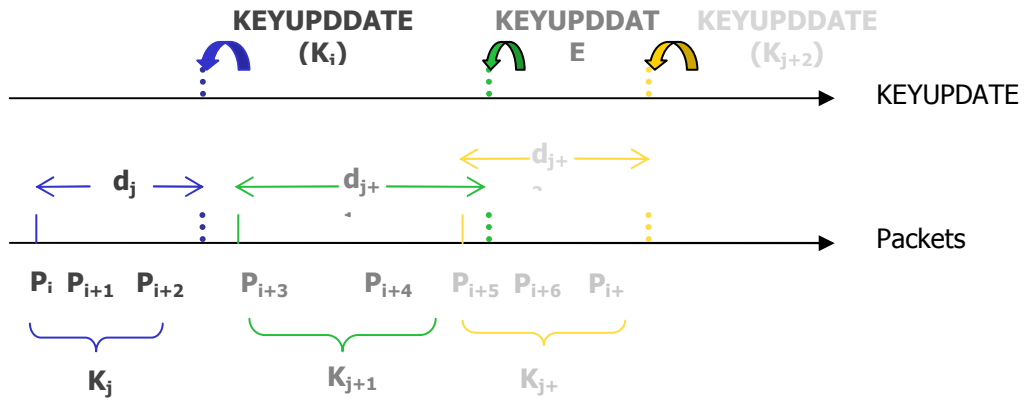


Figure 4. Varied delays of a key disclosure

3) Comparison with TESLA key disclosure scheme

The differences between our key disclosure and that of TESLA are:

- We broadcast KEYUPDATE message to release keys, while TESLA releases keys in data packets. Because different data packets may be targeted at different groups of receivers, TESLA is not able to guarantee that the key would be disclosed to all the receivers that have received the data packets protected by the key.

- Our protocol eliminates the need for clock synchronization. Clock synchronization has been proved to be prohibitively difficult and therefore we argue that it should be used in authentication mechanisms.

- In our protocol, the delay of key disclosure is not a fixed value since configuration of the network, as TESLA has used. It is up to the sender to decide the delay values based on the traffic status of the network. It allows more flexibility than TESLA and avoids the problem of authentication cache overflow.

3.2. Security Analysis

1) Trust management

Our protocol uses digital signature in both initial trust establishment and subsequent trust reestablishment. Compared to the scheme that uses asymmetric cryptography in only initial trust bootstrapping, our protocol can guarantee the genuineness of the key that commits to all the subsequent keys, and an “in-the-middle” attacker would not be able to replay an already released key and forge packets with the obsolete key afterward.

2) Message authentication

Up to date, MD5 [54] and SHA-1 [55] are two of the most widely used cryptographic hash functions. MD5 has been recently shown to be vulnerable to collision search attacks [56]. This type of attacks and other currently known weaknesses of MD5 can be thwarted by the use of MD5 within HMAC [57]. MD5-HMAC is proved to be more secure than MD5 in protecting the authenticity of traffic.

Our message authentication can effectively thwart the attacks of forging or maliciously alteration of packets.

3) Key disclosure

The delayed key disclosure can prevent from in-the-middle attack in which an adversary may use an obsolete key to forge or alter packets. However, the performance is dependent on the value of the delay.

Non-repudiation is also achievable in case of using large delay values.

3.3. Performance Analysis

In this section, we will evaluate the trust management and message authentication approach. We will evaluate the delayed key disclosure scheme as well.

3.3.1. Simulation setup

We use Network Simulator, ns2, for our simulations. We use two scenarios for our simulation:

Scenario 1: The first scenario we used is demonstrated in Figure 5. There are totally nine nodes in the scenario. Eight of them (denoted as N1, N2, … , N8 in Figure 5) serve as transmission nodes, who transmit packets to one single receiving node (denoted as N9 in Figure 5). Node N9 is the sink of all the traffic. The nodes are positioned at the mesh that is demonstrated in the figure. Static network topology used in this scenario allows us to easily observe the network performance (such as hop-by-hop delay, etc.) according to varied channel loads.

Scenario 2: In our second scenario, 50 mobile nodes are randomly distributed in a 1500x300 rectangular space. The node mobility model is random waypoint model, which is commonly used in simulations for mobile ad hoc networks. The maximum node speed is 20 m/s. This scenario allows us to observe the performance of our protocol in a

complicated environment that is more similar than Scenario 1 to a network in the real world.

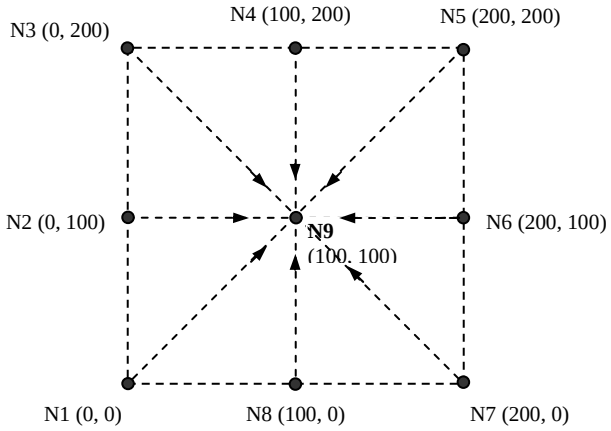


Figure 5. Network topology of a 9-node scenario

3.3.2. Performance evaluation for the trust management and message authentication

The performance metrics employed to analyze the trust management and message authentication approach are: *computational overhead*, *authentication latency*, *message overhead*. Our performance evaluation is based on theoretical analysis and simulation results.

We use a widely used simulation tool – *ns-2* [58]. Our simulation is based on a 1500 by 300 meters rectangle space. 50 nodes move from a random starting position to a random destination with a random speed uniformly distributed between 0 to 20 m/sec. If it is not specified, the pause time of nodes is set to 60 seconds. The Media Access Control layer protocol is IEEE 802.11 and the transport protocol is User Datagram Protocol (UDP),

which are both available as a part of the simulator. The length of data packet is 512 bytes and the traffic sources used are Constant-Bit-Rate (CBR).

1) Computational overhead

As any authentication mechanisms, our protocol introduces computational overhead by two operations: message authentication and trust management.

In our protocol, symmetric cryptography is used for message authentication. It is known that symmetric cryptographic operations are three to four orders of magnitude faster than asymmetric operations, especially on CPU limited devices.

We used asymmetric cryptography in trust bootstrapping, that is, when a node is establishing or reestablishing trust relationship with its neighboring nodes. This may introduce more overhead than LHAP because LHAP employs digital signature only when the trust is bootstrapped for the first time. However, we have argued that using digital signature is necessary even in re-bootstrapping since the key release is vulnerable to replay attack, especially when the receiving node has moved out of transmission range for a time interval and hence is likely to be unaware of the currently released key. It will not introduce significant overhead on receivers because signature verification is much faster than signature generation [59].

Moreover, our protocol only maintains one authentication key, which consumes fewer resources such as CPU and memory than LHAP, which maintains two keys – TRAFFIC key and TESLA key. We only use digital signature for trust bootstrapping. The trust maintenance is still based on one-way hash function, which is so efficient that it is usually considered negligible.

2) Authentication latency

The latency of authenticating a packet is introduced by two parts: MAC verification latency and key disclosure delay.

MAC verification is accomplished by computing one hash. The latency for this verification is less than one millisecond even for very constrained computational capability such as handheld PDAs [59].

The delay of key disclosure is a value that is determined by the sender of packets based on the traffic pattern. A very small delay may cause difficulty in satisfying the security condition and consequently increase the risk to key replay attack; while large delay may result in an increase on authentication latency. Tradeoff should be made between performance and security properties. A quantity analysis on the delay of key disclosure will be included in our future work.

3) Message overhead

Message overhead is introduced by trust management messages (such as trust bootstrapping, KEYUPDATE and trust relationship termination messages) and MACs of packets.

Suppose that the authentication is performed using MD5 Message Digest Algorithm. Then the MAC attached to each packet is a hashed digest that is 128-bit long. If the data packet size is 512 bytes, the overhead introduced by MACs is approximately 3%, which is very small.

The overhead introduced by trust management varies with the frequency of bootstrapping and KEYUPDATE messages. It is obvious that high node mobility will result in more frequent trust bootstrapping and therefore introduce more overhead. In

addition, a node sending more traffic will lead to more frequent broadcast of KEYUPDATE messages, which also introduces more overhead.

Figure 6 demonstrates the simulation results of the KEYUPDATE messages that have been resent. The data packet rates vary from 2 packets per second to 10 packets per second. We assume that the KEYUPDATE messages are sent with the same rate of the data packets. This implies that we use a new key for each data packet, which is the worst case for KEYUPDATE messages in term of message overhead.

We can tell that from the figure the packet resent rate increases with increase of packet rate. When data packets are sent with the rate of 2 packets per second, the resent rate is only 0.03%, which can be ignored. The resent rate increases to 37.33% when data packets are sent with a rate of 10 packets per second. In this case, the message overhead introduced by KEYUPDATE messages is 9.7% assuming that the identification of a node is128-bit long and the index of the key is 128-bit long too.

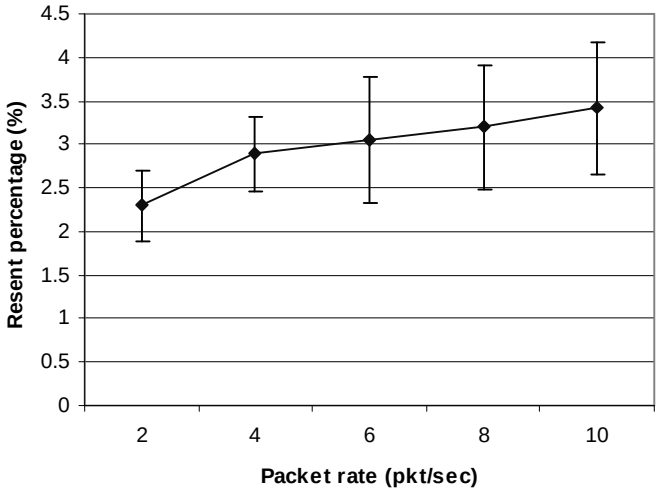


Figure 6. Resent rate of KEYUPDATE messages

The standard deviations of the percentages are shown in the figure too with the vertical lines on the values.

$$\sigma = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})^2}$$

Where σ denotes the standard deviation, $\bar{x}$ denotes the mean, and N denotes the number of the samples.

3.3.3. Performance analysis for delayed key disclosure

To analyze our delayed key disclosure scheme, we first take a measurement on average hop-by-hop delay. H*op-by-hop delay* of data packets is an important metric in determining the value of the delay that should be used in key disclosure scheme, in that the key disclosure delay should be large enough to guarantee arrival of the data packets before the key but meanwhile be as small as possible to achieve low authentication latency. We use hop-by-hop delay instead of end-to-end delay because our authentication protocol is designed for neighboring communications and the transmissions the protocol is aimed to protect are only one-hop transmissions.

Then we will use different key disclosure delay values to evaluate the performance, in metrics such as *percentage of packets arriving safely* and *dropped packet rate*.

1) Average hop-by-hop delay

We measured average hop-by-hop delay on both Physical Layer level and Network Layer. The delay on the Physical level is mostly the transmission time the packet takes in the air. We tested it in the scenario where there are two nodes, one of which transmits packets to the other. The distance between the two nodes is 150 meters. The average delay is 0.00467269 second with a deviation of less than 1×10^{-6} second.

The average hop-by-hop delay at the network layer is tested in both the scenarios of 9 nodes and 50 nodes we described earlier in this section. The hop-by-hop delay is calculated as *end-to-end delay* (a packet takes from the source to the destination) divided by the number of links a packet has traversed during delivery from the source to destination (the number of hops), i.e.

$$hop-by-hop\ delay = \frac{end-to-end\ delay}{number\ of\ hops}$$

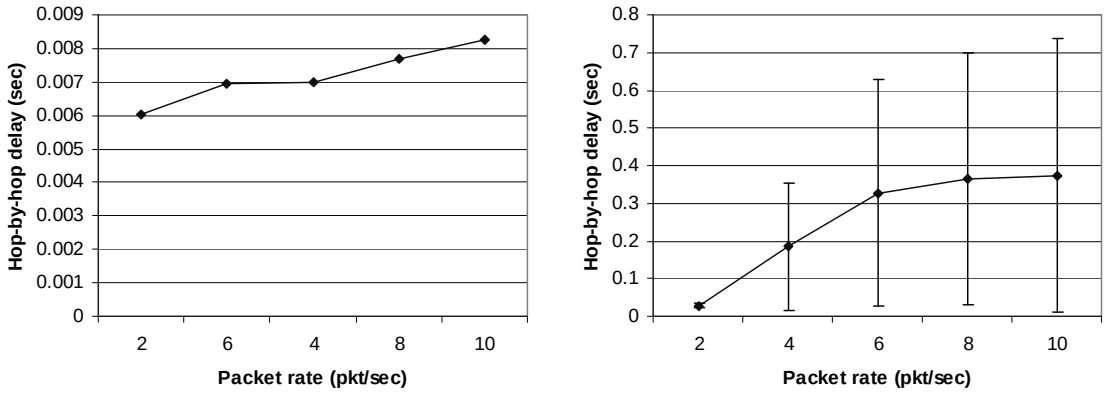
We measured the delay at the Network Layer because the key disclosure delay value (denoted as r in previous sections) will be determined and stamped on data packets above the Medium Access Control Layer level. Above Medium Access Control Layer, data packet delay may result not only from the transmission in the air but also from the *backoff* due to channel contention at Medium Access Control layer and from the *queue delay*.

The results for Scenario 1 (9 nodes) are shown in Figure 7 (a). The deviations are too small (less than 0.00002 second for all the cases) to be shown in the figure.

We tested average hop-by-hop delay in Scenario 2 with varied pause time, which is changed from 60 seconds to 480 seconds in an interval of 60 seconds (see Figure 7 (b)). The hop-by-hop delay for each packet rate is the average value from the cases with different pause time. The vertical line at each point presents the *deviation* of the value.

We can see from Figure 7 that average hop-by-hop delay increases with the increase of the data packet rate. The reason for this increasing delay is that increased packet rates result in larger channel load and therefore more channel contention for packets, and the

channel contention causes more backoff time for data packets. Table 2 and Table 3 give the average channel loads[4] according to the packet rates in Scenario 1 and 2 respectively.



(a) Hop-by-hop delay in Scenario 1

(b) Hop-by-hop delay in Scenario 2

Figure 7. Average hop-by-hop packet delay

With the same data packet rate, the average channel loads in Scenario 2 are less than the corresponding channel loads in Scenario 1. However, the delay values are larger than those in the scenario of 9 nodes. This is caused by the following reasons:

First, channel loads do not always accurately reflect the contention status at a channel, because a node's neighboring communications may also affect its capability of receiving packets and the packets in these communications are not accounted as its channel load. In Scenario 2, although the channel loads are lighter, the contention is more intensive in that most of the nodes have more neighbors than node N9 in the first scenario. As we have mentioned earlier, more intensive contentions result in more backoffs and hence larger transmission delays.

[4] Please note that here "channel" refers to the medium that a nodes shares with all its neighbors, which is different from "link", which refers to the point-to-point medium that two neighboring nodes use for transmission.

Second, node mobility may also introduce delays since it can cause re-routing when the network topology changes. These routing packets will compete with data packets for the bandwidth of channels and therefore cause more backoffs on data packets.

Table 2. Average Channel Load (Scenario 1)

Packet rate (pkt/sec)	Channel load (bps)	Channel load Percentage (%)
2	173974	8.70
4	374926	18.75
6	534254	26.71
8	733054	36.65
10	907016	45.35

Table 3. Average Channel Load (Scenario 2)

Packet rate (pkt/sec)	Channel load (bps)	Channel load Percentage (%)
2	57096	2.85
4	107592	5.38
6	125818	6.29
8	156477	7.82
10	182633	9.13

From the above simulations, we can conclude that *hop-by-hop delay increases with increase of traffic load in the neighborhood*. Therefore, a sender should use larger key disclosure delay in case of heavier traffic load.

2) Percentage of packets arriving safely

According to the average hop-by-hop delay demonstrated in Figure 7, we tested our key disclosure scheme with varied disclosure delay values. The percentages of data packets that arrive safely according to different data packet rates are shown in Figure 8. We observe that more than 97.6% of the data packets have arrived safely when the key

disclosure delay is set to 3 seconds; more than 94.8% of the data packets have arrived safely if the key disclosure delay is set to 2 seconds, in all the cases of different data packet rates.

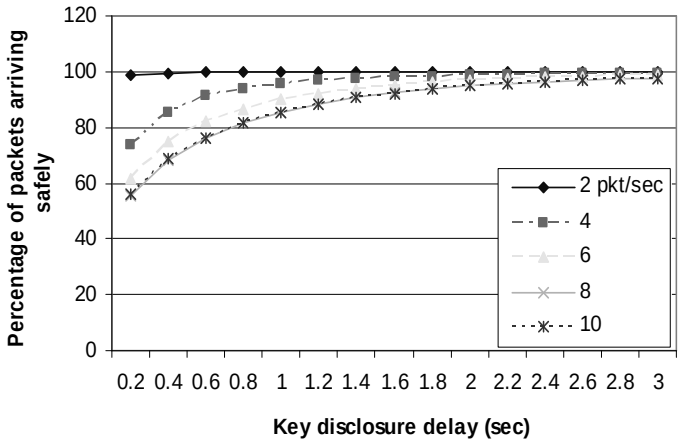


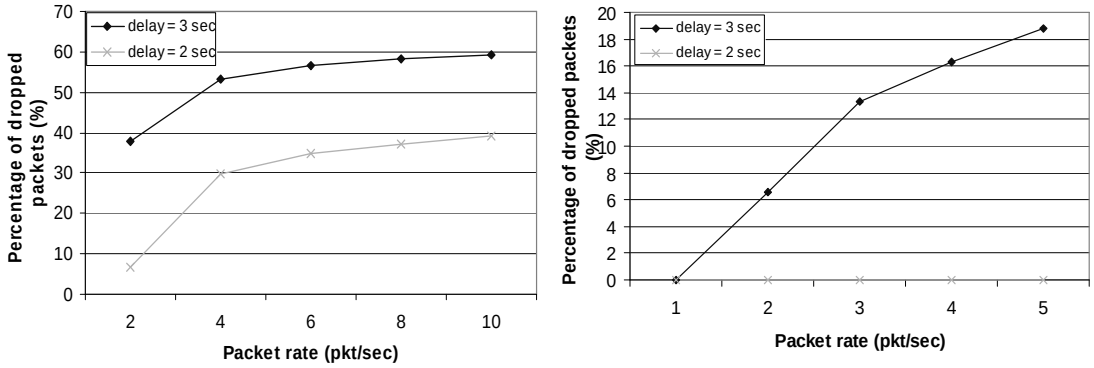
Figure 8. Rate of packets arriving safely

3) Dropped packet rate

We also test the dropped packet rates with different cache sizes. We use very small cache sizes (16 packets and 32 packets) to observe the performance. The key disclosure scheme should have less dropped packet rates in real networks since larger cache sizes (such as 128 packets) are often used.

The results for the two cache sizes are shown in Figure 9 (a) and Figure 9 (b) respectively. From the simulation, we have noticed that with increase of data packet rates, the drop rate at the cache increases too. We can also observe that, if the cache size is as small as 16 packets, there will be about 39% data packets dropped at the cache at 10 packets per second of data packet rate if the key disclosure delay is set to 2 seconds. In case of 3 seconds key disclosure delay, the drop rate will increase to 60% or so. With the cache size of 32 packets, drop rate decreases to 0 in case of 2 seconds or lower key

disclosure delay in case that the data packet rate is 10 packets per second. If the key disclosure delay is 3 seconds, the drop rate is about 19%. However, if we use a cache with size of 64 packets, the drop rate will drop to 0 no matter what the data packet rate is (in a 2 pkt/s to 10 pkt/s range).



(a) Dropped packet rate (cache size: 16 packets) (b) Dropped packet rate (cache size: 32 packets)

Figure 9. Average dropped packet rate

If we use a cache of 64-byte length, the dropped packet rate will be 0 even with 10 pkt/sec data packet rate.

3.4. Conclusion

Most ad hoc networks do not employ any network access control, leaving them vulnerable to resource consumption attacks. In ad hoc networks, users need to assure the party who supposedly sent a message to another party that it is indeed the legitimate party. Otherwise, a malicious node could tamper a network with falsified data. These attacks can result in degraded performance of networks, interference of resource reservation, and unauthorized use of resources. To deal with these attacks, an authentication protocol needs to be in place to ensure that a packet is sent by an authentic and legitimate node.

In this chapter, we have proposed a lightweight authentication protocol that effectively and efficiently provides security properties such as authenticity and integrity for communicating neighbor nodes in MANETs. The protocol utilizes *one-way hash chains* to compute authentication keys, which not only eliminates the high performance overhead imposed by *asymmetric* cryptography (such as digital signatures), but also avoids the difficulty of key management introduced by secret paired *symmetric* key. Our protocol also used *delayed key disclosure* to prevent a malicious entity from forging packets with Message Authentication Codes (MACs) with an already released key.

The authentication protocol is lightweight, scalable and tolerant of packet loss. The performance analysis showed that the protocol incurs low overhead penalty and also achieves a tradeoff between security and performance. The delayed key disclosure approach can achieve an extremely low dropped packet rate if the data packets are cached in a fair size buffer before being authenticated.

4. SECURITY IN QOS MODELS AND SIGNALING SYSTEMS FOR MOBILE AD HOC NETWORKS

Integrated Services (IntServ) [1] and Differentiated Services (DiffServ or DS) [2] are two commonly used QoS models that have been proposed for the traditional Internet and are also being investigated for MANET environment. Flexible QoS Model for MANETs (FQMM) [2] is a model proposed solely for mobile ad hoc networks. These QoS models specify architectures in which some kinds of services could be provided.

QoS signaling is used to search for routes with sufficient resources for desired QoS, to reserve and release resources, to set up, tear down and renegotiate flows in the networks. Some QoS signaling systems have been proposed for MANETs, such as INSIGNIA system [10], QoS AODV [11] and etc [14].

To solve the security problems of QoS signaling for MANETs, we propose a security mechanism for QoS Signaling Systems to provide authentication and detect malicious attacks on QoS parameters. We report our simulation results to demonstrate the low delay penalty achieved by the proposed system.

Security is a significant aspect for QoS signaling systems. However, there is little work published on the topic of intrusion detection and security prevention on QoS signaling.

While the two mechanisms that have been proposed, SDS/CD and RSVP-SQoS, protect the RSVP messages in an efficient and flexible manner, neither of them can be applied to MANET QoS signaling systems due to the following reasons:

- They employed digital signature mechanism for integrity and non-repudiation protection, which has been proved very expensive for MANETs in terms of message overhead and the computing complexity.

- They can efficiently detect misbehavior on concave QoS parameters such as bandwidth, but not applicable to additive metrics such as delay or jitter.

Therefore, to provide security characteristic to QoS signaling in MANETs, a new mechanism is necessary.

4.1. QoS Model Security in MANETs

The IntServ model provides an end-to-end QoS guarantee on a per-flow basis. It requires that every IntServ-enabled router keep the flow-specific states including bandwidth requirements, delay bound and cost of the flow, and therefore is not scalable for the Internet. However, the scalability problem of the Internet IntServ model is less likely to occur in the current MANETs in consideration of the small number of traffic flows and the limit size of the network [3]. In addition, because rapid change of nodal roles necessitates inclusion of all functions at all nodes in MANETs, the requirement that each node in the IntServ domain has to apply all the functions such as classification, admission control and scheduling, which deters the IntServ implementation for wired networks, does not introduce any extra problem for MANETs.

DiffServ is designed to provide more scalability and greater flexibility than the IntServ for wired networks. The DiffServ model is based on flow aggregation by classifying packets into a limited number of classes and then applying specific forwarding treatment to each QoS class. At the boundary of a DiffServ-enabled domain, the edge routers control the traffic entering the network with classification, marking, policing and shaping mechanisms.

Flexible QoS Model for MANETs (FQMM) [3] is a model proposed solely for mobile ad hoc networks. The FQMM takes the characteristics of MANETs into account and is a

hybrid provisioning scheme of the per-flow service in IntServ and the per-class service in DiffServ.

While the scalability and flexibility problems in QoS models have drawn extensive attention, there has been little work published in the aspect of security - another significant issue in MANET QoS models.

The characteristics of ad hoc networks such as exposure to hostile environment (e.g. battle field, rescue missions) and difficulty of authentication exacerbate the QoS model security problems. Without the protection of security mechanisms, a QoS model is vulnerable to both theft of service and denial of service, which inhibits the guarantee of network resources availability.

We discuss security issues of the three MANET QoS models.

4.1.1. DiffServ security in MANETs

Several vulnerabilities in DiffServ for MANETs make it a less secure model than the IntServ.

First of all, the DiffServ model is based on the trust relationship between edge routers and core routers for each DiffServ domain. Functions such as classification, marking, policing and shaping are all accomplished at edge routers where the flow enters the DiffServ network, while the core routers only forward the packets according to the service level marked in the Differentiated Service CodePoint (DSCP) field. As a result, if an edge router is compromised and makes malicious alteration on flows, the core routers can not find the on-going attacks since they are neither aware of the flow states nor do they have the capability of checking the correct DSCP settings in the packets.

This assumption of trust relationship is reasonable for the traditional Internet because a *security domain* can be established for each DiffServ domain, where core routers can therefore trust the edge routers. A *security domain* is ‘a set of machines under common administrative control, with a common security policy and security level. Hosts in this domain place a certain level of trust in the other hosts and may thus provide certain services for these trusted hosts which are not available to hosts residing outside of the security domain’ [11] . A DiffServ model can take advantage of this trust relationship to assure a certain level of security in the DiffServ domain.

However, the situation is different for MANETs. No third party is trustworthy in wireless ad hoc networks due to the fact that there is no fixed topology and therefore it is difficult and in some circumstances even impossible to establish a security domain in MANET environment.

The second vulnerability of the DiffServ model results from the ambiguous definitions of edge and core routers for MANETs. Some researchers proposed an architecture in which the sending node itself also performs as the ingress edge router and the destination node as the egress router [3]. This scheme allows a malicious sender trusted by other nodes who does not respect the QoS policy to be able to use as much resources as available.

Third, the absence of authorization facilities in ad hoc networks impedes the establishment of another line of defense. Because there is no central Policy Decision Point (PDP) (e.g. Bandwidth Broker) for the edge routers to consult in a MANET DiffServ domain, routers applying incorrect policy can have both unintentional and deliberate misbehaviors.

By exploiting the vulnerabilities described above, adversarial nodes can issue attacks in two ways.

First, illegal promotion of *Per-Hop-Behaviors* (PHBs), namely a base set of packet forwarding rules indicated by the DSCP in the IP packet header, can be accomplished by mis-marking the packet or shaping/policing a flow incorrectly at the ingress edge router.

Second, adversaries can steal or deplete the network resources legitimately reserved for other users via IP source spoofing. This form of attack issued in MANETs is more deceiving than that in the Internet. Figure 10 illustrates this case.

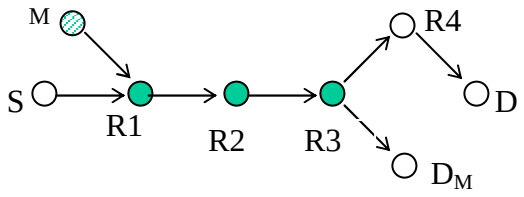


Figure 10. An example of theft of service in Diffserv model for MANETs

S is sending packets to node D through node R1, R2, R3 and R4. S has legally reserved 30% bandwidth over the path. Node M is a malicious node who successfully spoofs S's IP address and sends packets through R1, R2, R3 to D_M. If M marks its packets with the same DSCP value as the packets from S, it could use the reserved bandwidth for S at router R1, R2 and R3. Since routers forward packets based on aggregated traffic rather than on flows, R1, R2 and R3 would not even notice that the traffic from M is not destined to node D. Therefore, node M successfully steals bandwidth from node S, and can also affect other traffic at the three core routers.

Besides theft of service, denial of service (DoS) is also a major security risk to ad hoc DiffServ networks. DoS on QoS models could be a complete theft of service that is launched to penalize legitimate traffic. Similar to theft of service, it can be issued by means of IP source spoofing, inappropriate packet marking or erroneous flow policing.

4.1.2. IntServ and FQMM security in MANETs

Compared to the DiffServ model, the IntServ approach does not have the security risks mentioned above because it is based on flows rather than on aggregated traffic; classification, packet scheduling and admission controls are enforced at each router on the path according to the applicable policy, which eliminates the requirement of trust relationship among routers as well as the necessity of the central Policy Decision Point. Therefore, the IntServ avoids the vulnerabilities in the DiffServ model.

However, IntServ model requires a signaling system to achieve QoS provision along a transmit path. Without protection of certain security mechanisms, a QoS signaling system could become the target of malicious attacks. We discuss the signaling security in details in the next section.

The FQMM is particularly aimed at MANETs. It tries to take advantage of both the per-flow service granularity in IntServ and the service differentiation in DiffServ model and hence inherits the security vulnerabilities of both the IntServ and the DiffServ approaches.

4.2. Security Requirements and Attack Models for QoS Signaling Systems in MANETs

The most concerned security issues for QoS signaling systems include *integrity* of the signaling packets and *genuineness* of the network information. In spite of the fact that the network state information could be inaccurate in MANETs due to the node mobility and rapid topology change, a deliberate distribution of false information will lead to more disastrous results. In this section, we analyze the security requirements and attack models for QoS signaling systems.

4.2.1. Security requirements for QoS signaling systems

Without protection from a security mechanism, attacks on QoS signaling system could result in QoS routing malfunction, interference of resource reservation, or even failure of QoS provision. The security requirements for QoS signaling systems are as follows:

First, an integrity protection mechanism should be in place to guarantee that the *non-mutable* part in the QoS object, such as the QoS profiles for traffic flows, is not changed illegally. Illegitimate increase on QoS profile parameters could lead to unnecessary reservation for network resources or even failure of reservation in case that the network cannot accommodate the amount of service requested incorrectly; while decrease on the QoS parameters would affect the QoS provided to the flows because the reserved resource might be insufficient for the desired service.

Second, QoS states collected over the path should be resistant to attacks. The corresponding QoS parameters (e.g. available bandwidth and accumulative delay over a path) measuring these states are *mutable* at intermediate nodes. The malicious attacks on these parts are more deceiving than those on non-mutable parameters because they cannot be detected via integrity verification.

For example, in QoS AODV signaling, to determine whether a path can meet the required *Maximum Delay* specification of the QoS data, an intermediate node must compare its NODE_TRAVERSAL_TIME to the remaining *delay* indicated in the *Maximum Delay Extension*. If the *delay* is less than the NODE_TRAVERSAL_TIME, the node must discard the RREQ without processing it any further. Otherwise, the node subtracts the NODE_TRAVERSAL_TIME from the *delay* value and continues processing the RREQ. Therefore, the value of the Maximum Permissible Delay field should be decreasing during delivery of the RREQ packets, and likewise the values of the Maximum Permissible Jitter and Minimum Available Bandwidth should be decreasing as well.

A mistaken or malicious increase on these values would result in distribution of false network state information over the traversed path. A path with insufficient network resources could be established and the reservation would finally fail. An attacker who wants to disrupt the reservation could decrease the values by an extremely large amount, which however would only help the flows to avoid the malicious node. We will not deal with this situation in this work.

At last, network resources should be reserved correctly at each node along the path. A node should not be able to maliciously break the promise it has made of reserving the desired service without being noticed.

4.2.2. Attack models for QoS signaling systems in MANETs

We consider four attack models for QoS signaling system.

Attack model 1: Signaling message spoofing. An adversary can spoof signaling messages to request QoS, reserve resources or release resources. Falsified signaling messages can be used by illegitimate entity to steal resources, disrupt QoS services, which would consequently degrade the network performance. For example, a malicious node M spoofs signaling messages using node A's identification to reserve some resources. Node M can use these resources to transmit its own traffic (theft of services); or it can simply leave these resources unused so that the resources will not be available to other nodes (disruption of services).

Attack model 2: Denial of QoS request. An adversary can potentially intercept or drop reservation messages so that the QoS reservation and the channel setup will be failed or tremendously delayed. This attack can prohibit the QoS resources from being available to the victim.

Attack model 3: Malicious alteration of non-mutable parameters in transmission. For

example, an attacker can change the requested QoS in RREQ packets. It can also maliciously alter the QoS reservation parameters in RREP which will result in reservation of an incorrect amount of QoS resources.

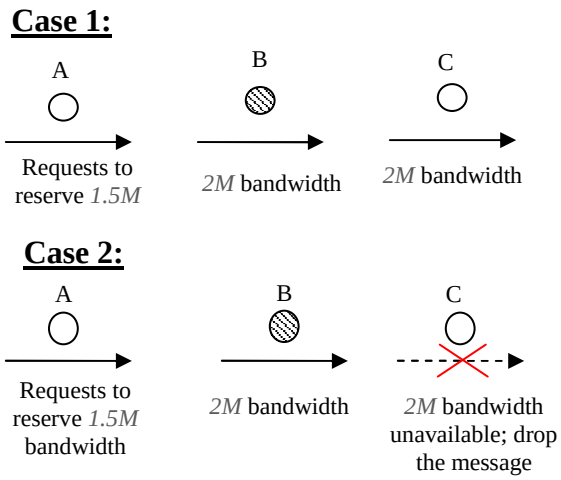


Figure 11. An example of malicious alteration of non-mutable parameters

Figure 11 is an example of this attack: node A receives a signaling message from originator S to request a reservation of 1.5M bandwidth. Node B is an adversary residing adjacent to A on the route who maliciously alters the request for bandwidth to 2M, which is larger than the original request value. If the attack is successful, the downstream nodes would not be aware of the malicious alteration. Therefore they would reserve 2M bandwidth in case that there is 2M bandwidth available at each downstream node (case 1 in the figure); or some downstream node will drop the request message in case it cannot provide 2M bandwidth (case 2 in the figure), even if it is capable of providing 1.5M.

If a malicious node decreases the value of the requested resources, it can result in a reservation of insufficient resources which can also disrupt the quality of the service provided to the flow from originator S.

Attack model 4: Intentional provision of fallacious QoS states information. Although QoS states information is subject to errors due to the rapid topology change and high node mobility, a deliberate distribution of false information will do more harm to QoS provisions. In this type of attacks, an adversary may tamper with the mutable QoS parameters (such as) in signaling messages in order to disrupt the measurement of QoS state and provide false information. The attacks may result in failure of resource reservation, insufficient or excess reservation.

Figure 12 is an example of this type of attacks on QoS AODV messages. Originator S sends a QoS request for 60 milliseconds (ms) delay. The *Maximum Permissible Delay* (MPD) parameter in the message is used to measure available delay along a candidate route. The original value of MPD is the requested delay and it should be decreasing downstream along the route. When the message reaches node A, whose traversal time is 25ms for example, A changes the value MPD parameter from 60 to 35. Suppose node B is a malicious node adjacent to node A on the route. Node B is supposed to deduct its own value from 35, but instead it increases the value of the parameter to 50ms. This may result in successful reservation along the route even if the route can not satisfy the QoS request of 60ms delay. In this case, the request of originator S would not be satisfied and the service is disrupted.

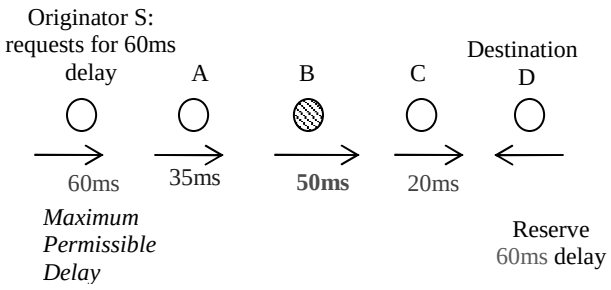


Figure 12. An example of intentional provision of fallacious QoS states information

It can be obviously seen that a QoS signaling system is vulnerable to various attacks without protection of security mechanisms.

4.3. Security Mechanism for QoS Signaling Systems in MANETs

In wireless ad hoc networks, QoS signaling is likely to be embedded with the routing protocols. Secure routing without QoS requirements is not within the scope of this work. Because secure routing protocols can be used in our scheme with suitable modification, we assume in this dissertation that the routing protocols are reliable and resistant to malicious attacks.

4.3.1. Hop-by-hop authentication protocol

In QoS-enabled ad hoc networks, users need to assure the party who sent a signaling message is indeed the legitimate party. Otherwise, a malicious node can tamper QoS signaling messages with falsified data to steal or deplete resources used or reserved by other nodes. These attacks can result in degraded performance of networks, interference of resource reservation, unauthorized use of resources, or even failure of QoS provision. To thwart these attacks, an authentication protocol needs to be in place to ensure that the originator of a packet is the authentic and legitimate node. An authentication protocol

should be lightweight and impose as small computational and message overhead as possible due to the fact that resources in a mobile ad hoc network are very limited.

The protocol described in the previous section is a lightweight hop-by-hop authentication protocol. It utilizes *one-way hash chains* to compute authentication keys, which not only eliminates the high performance overhead imposed by *asymmetric* cryptography (such as digital signatures), but also avoids the difficulty of key management introduced by secret paired *symmetric* key. To generate a key chain of length n+1 in a one-way hash chain authentication, the first element of the chain (denoted as h_n) is randomly picked and then the chain is generated by repeatedly applying a one-way function H (h_n, h_{n-1}, ..., h_0). A one-way hash function maps an input of any length to a fixed-length bit string, which is defined as $H:\{0,1\}^* \rightarrow \{0,1\}^\phi$, where ϕ is the length of the output of the hash function – the newly generated key. In utilization and revelation of these keys, the reverse direction of key generation is used: start from h_0, the last generated key, and then h_1, ..., h_{n-1} in sequence.

When a node wants to send a message, it computes the MAC on the message and then unicast to the receiving node (say node B), or multicast (or broadcast) the packet (denoted as P^A) to the receivers in the following format:

$$A \rightarrow B(*): M, MAC(M, h_i^A)$$

where h_i^A is the currently used key of node A's. Note that the key h_i^A has not been disclosed at this point. The originator of the packet (node A in this case) will later disclose h_i^A in a KEYUPDATE message. The key enables the receiver to verify the MAC of the message. If the verification is successful, the message is then authenticated and trusted. Once the key is disclosed, it becomes obsolete and can not be used to generate MACs any more.

This authentication protocol will be used to protect the authenticity of QoS signaling messages hop-by-hop.

4.3.2. Basic scheme of the security mechanism for QoS signaling systems

We use *end-to-end* authentication for the non-mutable parameters in QoS signaling messages. Our approach requires the originator or the destination node to digitally sign the *non-mutable* parts of the QoS AODV packets, such as the QoS profile of the flow from the originator or the reservation request from the destination.

Before sending a RREQ message, the originator signs the QoS parameters with its private key. Each intermediate node on the path can *voluntarily* verify the digital signature to assure that the QoS parameters have not been maliciously altered during transmission. After the RREQ reaches the destination node, the destination checks the integrity of the non-mutable QoS objects via MAC verification. If the objects have been altered during transmission, the destination node will raise an alarm. Otherwise, it generates RREP packet, hashes the QoS parameters and sends it back to the originator of the request. The originator will verify the authentication and integrity of the QoS parameters upon receiving the RREQ packet from the destination.

For the *mutable* parameters, we will use the *hop-by-hop* authentication protocol described in previous section as our authentication mechanism. Each intermediate node generates MACs with its currently used hash chain key and then relays the RREQ packet to its adjacent downstream node. After the key is disclosed with a delay since the packet has been sent, the downstream node will use the disclosed key to verify authenticity and integrity of the parameters. In case that the authentication fails, the node will raise an intrusion alarm to its downstream node on the path as well as all the other neighbors. This mechanism can prevent spoofing signaling messages and protect legitimate signaling messages from in-the-middle attack.

To prevent from intentional provision of fallacious QoS states information as exemplified in Figure 12, we use a mechanism that works in a similar way to *watchdog* [60], which was proposed to detect routing misbehavior in mobile ad hoc networks. Our mechanism requires that each intermediate node on the route send a signaling message not only to its downstream neighbor, but also to all the other neighbors. That is, an intermediate node is required to broadcast the signaling message instead of unicasting to the downstream node. The upstream node will listen to the broadcast signaling message and verify if its neighbor is maliciously distributing false QoS status.

Figure 13 is an example of our intrusion detection scheme. Suppose there exists a path between originator S and destination D. Nodes A, B and C are intermediate nodes on the route. S wants to send a flow that requires a delay of less than 10 milliseconds and therefore sends a RREQ message with value of 10 milliseconds for Maximum Permissible Delay parameter. When S initiates the request, it adds the MAC of the Maximum Permissible Delay, which is denoted as M_s in Figure 13. When node A receives the RREQ packet, it calculates the new *delay* value and appends the value with its MAC of the Maximum Permissible Delay field. Node A will then broadcast the value with its MAC to its neighbors so that node S will be able to receive the message and verify if the value is reasonable. For example, if the NODE_TRAVERSAL_TIME at A is 2 milliseconds, the new *delay* value sent by A should then be 8 milliseconds. If A sends a value that is apparently invalid (such as 10 or larger), node S will raise an intrusion alarm. Both S and B will be able to authenticate the message using the later disclosed key.

Now we assume node B is a malicious node that is seeking chance to disrupt QoS provision. If it raised the *delay* value from 8 to 12, node A should be able to find out the *delay* has been increased by overhearing B's signaling message to C.

Our mechanism is also applicable to the Maximum Permissible Jitter and the Minimum Available Bandwidth fields.

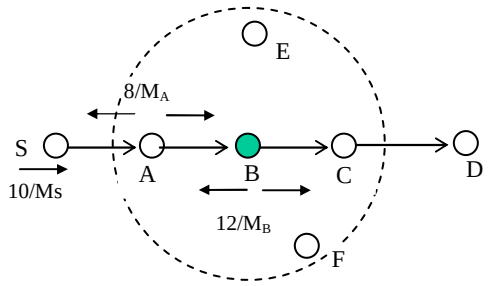


Figure 13. An example of intrusion detection for QoS signaling systems

To reduce the delay that our system may impose on the routing, the authentication and verification of the QoS values can be achieved offline. That is, an intermediate node can forward the RREQ first before it performs the security verification.

Under the circumstance that a node experiences a significant change that keeps it from reserving the promised service, it will send an *ICMP QOS_LOST* message. This could be helpful to the observation on a node's behavior by keeping a record for the nodes who have sent the *QOS_LOST* messages, and therefore help to detect malicious attack.

To prevent a malicious node from acting normal during the QoS signaling but failing to keep the promise intentionally, the destination node and volunteer intermediate nodes should monitor the flows against the promised QoS level and periodically report to other nodes including the originator of the flow.

4.3.3. Enhanced scheme of the security mechanism

Although the "Watchdog" scheme provides prevention for the integrity and authenticity of signaling messages and is capable of detecting intentional distribution of false QoS states, it is still subject to attacks. If a malicious node (say node B) intentionally sends false QoS status to the downstream node C when node A's radio channel is busy in order

to cause a signal conflict at A, then A will not be able to overhear the fallacious information. Later B sends the true information to node A while taking advantage of signal conflicts at node C so that node C would not be able to detect that node B sent different values. In this case, node A will fail to detect the fallacious QoS information distribution. Figure 14 exemplifies this case.

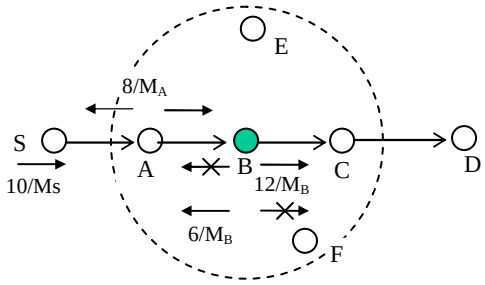


Figure 14. An example of intrusion on our security mechanism

To solve the problem, we have neighbors of an intermediate node on a data path participate in the detection (node E and F in Figure 14). Neighbors E and F are likely to hear both two broadcast signaling messages and therefore able to detect B's misbehavior (illustrated in Figure 15). The neighbors could cooperate in the detection for their own benefit because:

- A misbehaving node is very likely to issue attacks to disrupt their service as well.

- In MANETs, bandwidth is reserved not only at the relay nodes along the route, but also at each relay node's neighbors. A disruption of QoS provision can also waste the neighboring nodes' resources.

- A credit system can be used to stimulate the cooperation in detection [61].

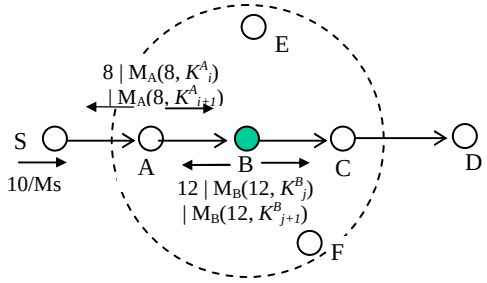


Figure 15. An example of cooperation of neighbors in our security mechanism

However, to have neighbors to join in the detection, a security mechanism that provides *non-repudiation* is required. For example, if node E or F detects misbehavior from node B, node E or F should give evidence that unambiguously shows that the attack was issued by node B and B is not being wrongfully accused. Our hop-by-hop authentication only provides non-repudiation of MACs when the key used to generate the MACs have not been released. That is, if an accusation of attack with a MAC happened before the key is released, the accused will not be able to deny it.

However, we do not want to delay the disclosure of the authentication key to a fairly long time later because otherwise the authentication of signaling messages would be also tremendously delayed. Moreover, the end-to-end delay of signaling messages would be significantly increased in case that a node wants to verify the authenticity before it relays the signaling messages. Therefore, we use two MACs for mutable parameters instead of just one: one is generated using the key that will be disclosed next; the other is generated with the key that will be disclosed after the next one. For example, in Figure 15, node A will send the Maximum Permissible Delay parameter, which is 8 ms in our example, with a MAC generated with key K_i^A (denoted by $M_A(8, K_i^A)$ in Figure 15) and a MAC generated with key K_{i+1}^A (denoted by $M_A(8, K_{i+1}^A)$). K_i^A is the key that is currently used and will be disclosed shortly; while K_{i+1}^A will not be disclosed until some time after K_i^A is

disclosed. Disclosure of K_i^A provides timely authentication for signaling messages, and delayed disclosure of K_{i+1}^A provides non-repudiation to neighbors' detection.

Suppose node B sends a signaling message with different Maximum Permissible Delay values to node A and C respectively. The misbehavior will be noticed by node E and/or F. Node E and/or F will raise an intrusion alarm with the two messages. The only way for B to deny the intrusion is to release the second key (K_{j+1}^B in our example) and proves that the second MAC of the message is not generated with K_{j+1}^B. K_{j+1}^B is authenticated by applying H_B to K_j^B. If node B can not prove it, the accusation is successful.

4.4. Security Analysis

Attack model 1: Signaling message spoofing. Our security mechanism uses digital signature to protect the authenticity of non-mutable parameters in the signaling messages (requested QoS by the originator or the reservation request by the destination).

We also designed a lightweight hop-by-hop authentication protocol to provide authenticity to the mutable parameters of signaling messages (measurement of available resources along a candidate route). Delayed key disclosure guarantees that a malicious node is not able to forge MACs with an already released key.

In our mechanism, as long as the key is not compromised, the identity of a legitimate node can not be spoofed by an attacker. In one-way hash chain authentication, we choose the function *H* that is simple to compute, nonetheless is computationally infeasible in general to invert. Therefore, it is extremely difficult, if not impossible, to guess the key based on the already released keys.

Attack model 2: Denial of QoS request. A malicious node may intentionally drop QoS requests from a specific node in order to prohibit QoS from being available to the victim. We use “overhearing” technique in signaling messages relay, therefore an upstream node is able to listen if the node has delivered the messages to another node. The upstream node may also be able to observe the adjacent node’s traffic and analyze if the drop is caused by insufficient resources or malicious intention.

Attack model 3: Malicious alteration of non-mutable parameters in transmission. By utilizing digital signature, the non-mutable parameters in QoS request or reservation messages can be effectively protected.

Attack model 4: Intentional provision of fallacious QoS states information. To thwart this type of attacks, we take advantage of the characteristics of open medium in MANETs in our intrusion detection mechanism. An upstream node can detect false QoS state information deliberately distributed by its adjacent downstream node. This hop-by-hop detection is not only able to detect attacks fast but also capable of locating the malicious node on the path, so that the malicious node can be punished or even excluded from the network to prevent further attacks.

In our security mechanism, two MACs are used to provide non-repudiation in case neighbors want to accuse some node. The approach is based on delayed key disclosure in order to prevent in-the-middle attacks. However, the value of the delay is yet to be studied to make the mechanism effective (to guarantee non-repudiation) as well as efficient (small detection overhead) in the detection.

4.5. Simulation Results

We built our simulation using Network Simulator *ns-2* [58]. The AODV simulation is part of the simulator. We added *delay* field according to the model proposed in [60] in

our simulation to serve as QoS field in AODV and then developed our Secure QoS Signaling system. We only tested the *delay* field because the protection of other fields such as *bandwidth* and *jitter* is the same as that of the *delay*. We use the MD5 Message Digest Algorithm [54] with protection from Keyed Hashing for Message Authentication [15] (MD5-HMAC) to generate the MACs. The MAC code from Black's publication [62] is also used in our simulation. We evaluated our system in this chapter based on the simulation results.

4.5.1. Simulation setup

Our simulation is based on a 1500 by 300 meters rectangle space. 50 nodes move from a random starting position to a random destination with a random speed uniformly distributed between 0 to 20 m/sec. The pause time is set to 600 seconds. The Media Access Control layer protocol is IEEE 802.11 and the transport protocol is User Datagram Protocol (UDP), which are available as a part of the simulator. The length of data packet is 512 bytes and the traffic sources used are Constant-Bit-Rate (CBR).

We initialized the delay requirement to 300ms, while the NODE_TRAVERSAL_TIME is set to 30ms as it is set by default in the AODV part of the simulator.

4.5.2. Performance evaluation

The performance metrics employed to evaluate our system are: *message overhead*, *average route request end-to-end delay, average security verification overhead* and *detection accuracy*.

1) Message overhead

This is the number of bits that our security mechanisms introduced on the RREQ packets.

If the basic security mechanism where cooperation of the neighboring nodes is not used, the message overhead introduced by the hashed digest in RREQ packets is 128 bits, which is 30.2% of the original QoS AODV packets; while in case that the cooperation of neighboring nodes is stimulated, the overhead introduced by the two hashed digest is 256-bits long, which is 60.4% of the QoS AODV RREQ packets.

2) Average route request hop-to-hop delay

It is the average of the delays incurred by all the route request packets that are successfully transmitted hop-by-hop. Because our hashing functions impose delay penalty mainly on route requests, we did not include data packets delivery delays into our metrics.

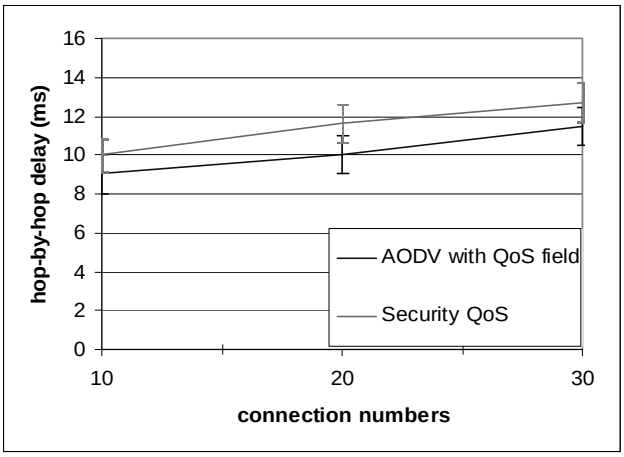
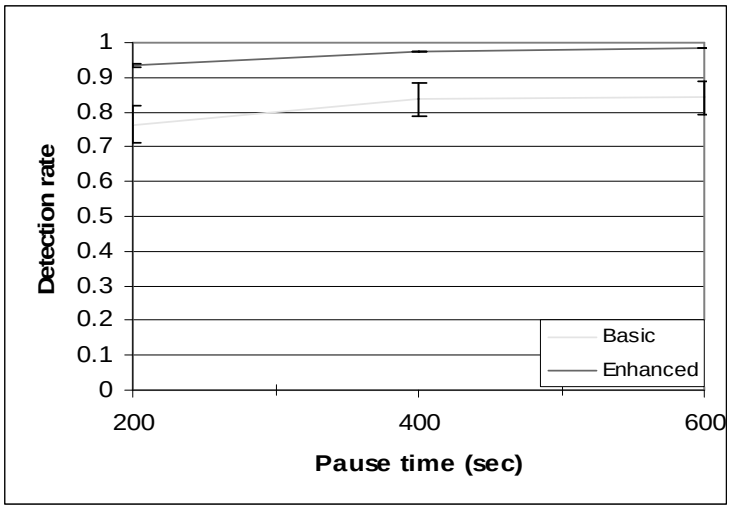


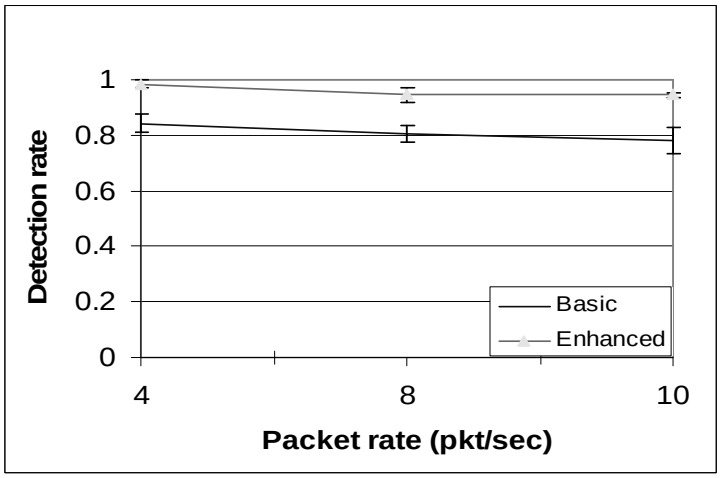
Figure 16. Average hop-by-hop delay of route request packets

We tested scenarios which include 10, 20 and 30 connections respectively. The results of the QoS AODV and our security protocol (without neighbor cooperation) are listed in Figure 16. From the figure we can see that the delay penalty imposed by our security

mechanism is negligible. We also noticed that the delay of our security approach with neighbors' cooperation does not impose any overhead to the delay compared to the basic security scheme. The reason for this is because computation of MAC is very fast (around 0.05ms) therefore it does not affect the route request delay.



(a) 50 nodes, 20 source nodes, rate: 4 pkt/sec



(b) 50 nodes, pause time: 600s, max speed: 20m/s

Figure 17. Intrusion detection rate for QoS signaling system

3) Average security verification overhead

This is the overhead that the security verification introduces, which will not be imposed on QoS signaling since it is accomplished offline. The security verification is completed approximately in 0.05325 milliseconds at each node in our simulation.

4) Detection rate

This is the ratio of successful detections over total number of misbehaviors. Our simulation results (as in Figure 17) show that our enhanced scheme achieves better detection rate than the basic scheme. Also, the detection rates increase with increase of pause time, while decrease with increase with data rate.

4.6. Conclusion

In this chapter, we addressed the security issues for MANET QoS signaling systems. Due to the nature of node mobility and the severe overhead imposed on the signaling systems, the existing countermeasures to attacks on QoS signaling for the traditional Internet cannot be applied to MANET environment. In order to detect misbehaviors on QoS signaling, we proposed a Secure Mechanism for QoS Signaling system. Our simulation results have demonstrated that the proposed good performance with high detection rate and low delay penalty.

5. INTRUSION DETECTION FOR BANDWIDTH RESERVATION IN MOBILE AD HOC NETWORKS

The security properties that should be supported for QoS in MANET include *availability*, *authenticity*, *integrity* and *confidentiality* [5]. *Availability* refers to the requirement that the service offered by a node should be available to its users when expected. It is a primary security property ensuring *soft* QoS provision in MANETs. *Authenticity* ensures the principals with whom one interacts are the expected nodes. *Integrity* enforces that a node or message transmitted has not been maliciously altered and *confidentiality* protects the secrecy of communication.

The properties of authenticity, integrity and confidentiality can be protected with existing approaches such as encryption and digital signature. However, new security mechanisms need to be designed to protect *availability* property in QoS systems. Without protection, QoS systems are vulnerable to various attacks such as *Denial of Service* (*DoS*) *attacks* and *QoS attacks*. DoS attacks can cause depletion of memory, CPU and network resources, and have been a serious threat for the Internet as well as for wireless networks. The aims of a QoS attack include theft of network resources (e.g. bandwidth) or degradation of the services perceived by users. Both DoS and QoS attacks may result in inaccessibility of network resources and therefore failure in QoS provision.

Several monitoring techniques have been proposed to detect QoS attacks or DoS attacks on QoS resources in the Internet [63]. To detect violations on QoS, Service Level Agreement (SLA) parameters such as *delay*, *loss* and *throughput* are monitored. Throughput measurement is to ensure that no user is consuming excessive bandwidth. In these approaches, once service violations are detected, the monitor will alarm for bandwidth theft or DoS attacks and then appropriate actions will be taken to eradicate the malicious nodes.

The approaches of monitoring delay and packet loss can also be used in mobile ad hoc networks to detect attacks on these SLA parameters. However, the technique for detection on bandwidth reservation (or monitoring throughput) can not be effectively applied in MANETs due to the unique characteristics of bandwidth reservation in MANETs.

MANETs are characterized by open shared medium, absence of fixed infrastructure and rapid topology change. These characteristics determine that providing security protection to bandwidth reservation in mobile ad hoc networks is very challenging and different from that in the traditional wired networks. First, any channel link of a node is shared with all its neighbors[5] in MANETs. That is, a node can successfully use the channel only when all its neighbors do not transmit and receive packets at the same time, which is termed as "aggregation effect" [15]. Therefore, to reserve bandwidth in MANETs, available bandwidth needs to be examined and reserved not only at forwarding nodes but also at their neighboring nodes. Many new approaches that analyze or implement bandwidth reservation have been proposed for mobile ad hoc networks [15] [64] [65]. Second, an intrusion can be detected in wired networks in case that the violation of bandwidth reservation has exceeded a predefined threshold. In MANETs, however, the communication capacity between any two nodes can be dramatically changed due to high node mobility, which may result in breaking previously promised bandwidth. That is, a violation of the agreement on bandwidth reservation may result from malicious attacks as well as non-malicious behaviors (such as a node wandering into the neighborhood without knowledge of the reservation, or signal interference from far transmission). For this reason, the intrusion detection mechanism for bandwidth reservation in MANETs should be able to differentiate misbehaviors from non-malicious behaviors.

[5] We define neighbors as the nodes that are within the communication range of a node and we assume bi-directional radio links in the network.

In this chapter, an intrusion detection mechanism will be proposed to detect malicious attacks on bandwidth reservation in MANETs. The aim of the detection mechanism is to ensure that the bandwidth reserved for a specific traffic flow would not be tampered with by a malicious node, who may violate the agreement on bandwidth reservation by intentionally preventing reserved bandwidth from being available.

The rest of this chapter is organized as follows: first we will give a description on bandwidth reservation in MANETs and the attack models on the reservation mechanisms; the intrusion detection mechanism we designed for bandwidth reservation will be discussed later; simulation results will also be demonstrated; then the chapter is concluded.

5.1. Bandwidth Reservation and Attack Models in MANETs

Before providing bandwidth guarantees for a traffic flow, the available bandwidth is first measured at each intermediate node. To determine whether there is enough bandwidth available for a new flow along a candidate data path, each intermediate node's available link capacity and the bandwidth to be consumed by the requesting flow should be measured. In the traditional Internet, bandwidth measurement is a trivial task because the underlying medium between any two nodes is a point-to-point link with fixed capability. However, the problem is complicated in mobile ad hoc networks due to the fact that communication links in MANETs are open medium and the radio channel of a node is shared with all its neighbors. In MANETs, a node can successfully use the channel only when all its neighbors do not transmit and receive packets at the same time. Under this effect, a node cannot use specific bandwidth simultaneously with any of its neighbors except for the receiving node, who will be listening to the channel during the transmission. Consequently, to determine whether there is sufficient resource for a QoS request in a mobile ad hoc network, a node needs to know its own available bandwidth as well as the available bandwidth at all its neighbors. Moreover, the bandwidth should

be reserved not only at each intermediate node (or forwarding node[6]) on a forwarding path, but also at all the neighbors of the intermediate node.

Figure 18 shows an example of aggregation effect in which a flow originated from node *S* traverses a forwarding node *F* and is destined to node *D*. When node *F* is receiving or transmitting packets, its neighbors *A*, *B*, *C* and *E* should remain silence because otherwise it may cause conflict, in which case the receiving and transmission would fail.

Suppose node *S* sends a QoS request for bandwidth reservation. The bandwidth should be reserved not only at *F*, but also at *A*, *B*, *C* and *E*, in that it needs to guarantee that there is sufficient bandwidth available at the entire neighborhood of *F* in order to provide successful bandwidth reservation.

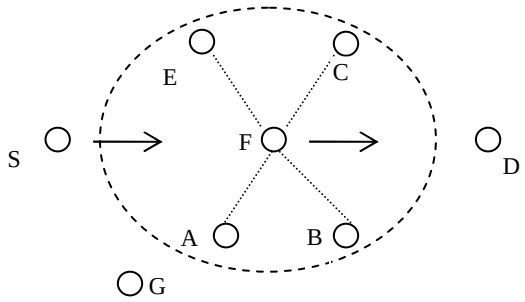


Figure 18. An example of aggregation effect in delivery of packets

In both MANETs and the Internet, bandwidth reservation is achieved by reserving specific time slots or sessions with the forwarding nodes (as well as neighbors of forwarding nodes' in case of mobile ad hoc networks). In this work, we do not assume any specific bandwidth reservation protocols on the Medium Access Control (MAC) layer or any specific routing protocols on the network layer. Our intrusion detection

[6] Henceforth, we will use the terms "intermediate nodes" and "forwarding nodes" interchangeably.

mechanism for bandwidth reservation can be an independent additional layer above the network layer, as illustrated in Figure 19.

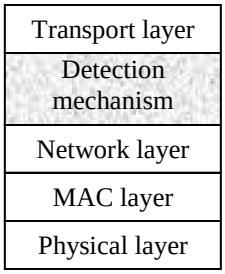


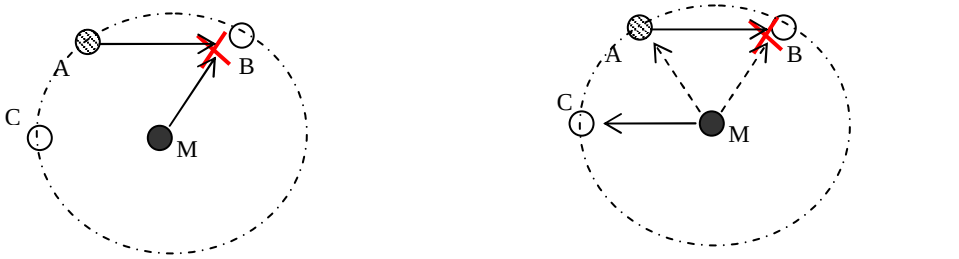
Figure 19. Integration of our detection mechanism in the OSI model

5.1.1. Attack models for bandwidth reservation in MANETs

Due to the characteristics of bandwidth reservation in MANETs, the reservation is vulnerable to various attacks. The attacks can be launched not only from forwarding nodes on the data path, but also from the neighbors of the forwarding nodes. Besides, because the medium is open and shared, malicious nodes do not even need to obtain physical access to a node or a channel to launch successful attacks. There are two attack models for bandwidth reservation in MANETs:

Attack model 1: DoQoS. Since bandwidth needs to be reserved at all the neighbors of a forwarding node, the neighbors should keep silence during the reserved time-slots or sessions once a reservation has been established. A malicious neighboring node may intentionally break previous reservation by transmitting signals during the reserved slot/session, which will cause *signal collision* and consequently failure of QoS provision. The objectives of this type of attack include disruption of bandwidth reservation (illustrated in Figure 20 (a)), or theft of reserved bandwidth to be used for other traffic

flows (in Figure 20 (b), M sending to C which causes conflict at B). Both these two attacks result in Denial of QoS (DoQoS) for the reserving flow.



(a) An example of disruption of bandwidth reservation attack

(b) An example of theft of bandwidth attack

Figure 20. An example of DoQoS attack in bandwidth reservation

It's worth mentioning that an attacker may send packets with falsified source address or simply transmit deteriorated packets or noisy signals so that other nodes would not be able to detect the source of the intrusion.

Attack model 2: QoS attacks. A selfish or malicious node on the data path may intentionally break the promise of bandwidth reservation. This type of attack includes two cases: (1) A malicious node may refuse to forward the packets during the reserved slots/sessions, simply leaving the bandwidth unused and wasted. The attacker may intend to disrupt the QoS, or may just be selfish trying to save its resources such as energy. (2) An attacker may use the bandwidth to deliver other flows such as its own traffic or traffic from which it would earn more profit. We call this type of attack as theft of bandwidth (or theft of service). This scenario is different from DoQoS attacks in that the packets received are characterized as undeteriorated.

5.2. Intrusion Detection on Bandwidth Reservation in MANETs

5.2.1. Assumptions

We assume that the network topology is fairly stable, or not changing so rapidly that may consequently cause failure of QoS provisioning. Specifically, if a node has promised QoS for a flow, it would not move out of the transmission range of its upstream and downstream nodes during the reserved time duration[7].

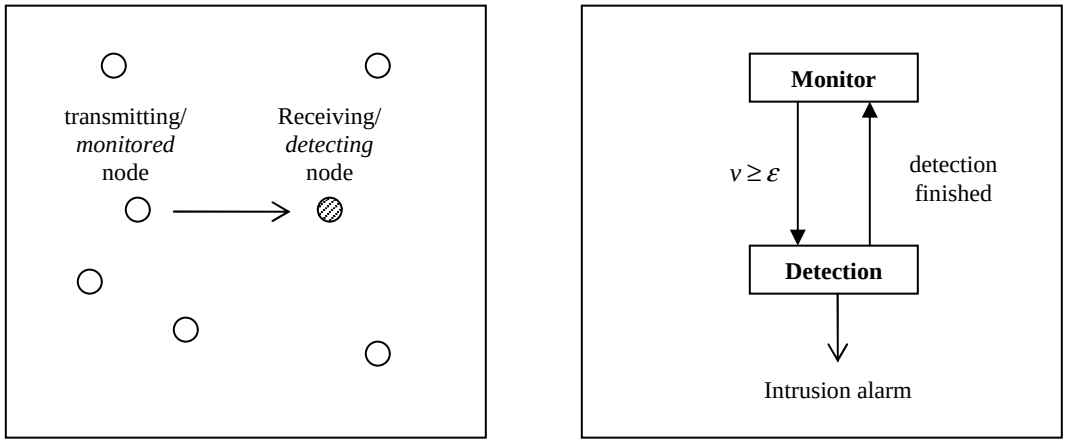
We assume that a node is able to estimate its current battery level and expected energy consumption before accepting a reservation. A benign node would not accept a QoS reservation if its energy level does not allow itself to provide the requested QoS during the reservation time duration. Therefore, we do not differentiate the case of exhausted battery from intrusion in our detection. It is also assumed that a node is able to predict its moving speed and estimate its position on the path. A benign forwarding node would not make a reservation if it is moving out of the transmission range of its upstream node and downstream node on the data path during the reserved duration. In reality, an unexpected event such as power failure due to mobility can happen to benign nodes. In such cases, reputation systems [66][67][68] can be used to evaluate the behaviors and to detect intrusions.

5.2.2. Intrusion detection mechanism

Our mechanism is composed of two modules: *monitor* module and *detection* module. We use *hop-by-hop* monitoring and detection. Specifically, the *receiving node* monitors and detects misbehaviors from the transmitting node. During the reserved time duration, the *monitor module* on a receiving node monitors the *throughput* with which its

[7] Reserved duration refers to the time period during which the reservation is valid; while the reserved slot or session refers to the time division that the transmission for the flow according to different MAC protocols.

upstream transmitting node has delivered the reserving flow (see Figure 21 (a)). Here we detect violations on bandwidth reservation by observing the transmitting node's throughput. The throughput is used to calculate the bandwidth that the transmitting node has used in transmitting the packets. If the bandwidth actually used is less than the reserved bandwidth and the violation exceeds a threshold (i.e. $v \geq \varepsilon$, where v denotes the violation and ε denotes the threshold), it will notify the *detection module* and the detection process will be launched. The aim of the *detection module* is to determine whether the violation is caused by malicious behaviors. DoQoS and QoS attacks will be differentiated in the detection. When the detection finishes, the status will return to "monitor" no matter whether an intrusion alarm has been issued or not, as illustrated in Figure 21 (b).



(a) Roles of nodes in Intrusion Detection

(b) Components of the detection mechanism: monitor and detection modules

Figure 21. Roles of the nodes and components of the detection mechanism

In our intrusion detection, QoS attacks are identified with upstream node sending other flows (theft of service) or received power R at the receiving node lower than a predefined threshold t_R (bandwidth unused); while DoQoS attacks are identified with the

signal strength received at the receiving node exceeding the estimated maximum interference but no new neighbor has been detected.

Figure 22 demonstrates the process of the intrusion detection. The procedure will return to the *monitor module* when it finishes.

```
Procedure IntrusionDetection ( )
Input: Received power R, minimum used bandwidth t_R, Estimated maximum
       interference E[I_max]
Output: Alarm of DoQoS or QoS attacks, or No alarm
Begin
  If undeteriorated packets for other flows received
   Then return QoS_Attack_Alarm
  Else
    If received power R <= t_R + λ1
     Then return QoS_Attack_Alarm
    Else
      If received power R >= E[I_max] + λ2
        If new neighbor(s) detected
         Then  Begin
                 Negotiate with the new neighbors
                 return no alarm
               End
         Else
           return DoQoS_Attack_Alarm
   return no alarm
End.
```

Figure 22. Pseudo code of the intrusion detection algorithm

In Figure 22, t_R denotes the threshold which is the minimum power needed to receive a flow with the reserved bandwidth in case of no signal transmission interference. If the transmitting node has not sent signals with the power strong enough to be received even when there is no interference, it is apparently issuing a QoS attack. We use the Shannon-Hartley formula to calculate the threshold t_R. The formula is:

$$C = B\log_2(1+S/N) = B\log_2(R/N) \qquad (1)$$

where C is the capacity in bits per second, B is the bandwidth of the channel in Hertz, and S/N is the signal-to-noise ratio. The capacity C is the theoretical maximum rate of clean data of the channel. If we let C be the *reserved bandwidth*, B be the *raw data rate* of the channel, and N_t be the *thermal noise* in the environment, then R will be the minimum power threshold that is needed to transmit the flow, i.e. $t_R = N_t * 2^{B_{rsvd}/B_{raw}}$.

E[I_{max}] denotes the estimated maximum interference in a benign environment without denial of service. If the signal interference calculated, based on measurement of the throughput, is larger than E[I_{max}], we can conclude that there is a DoQoS attack.

λ_1 and λ_2 are the security factors used to adjust the detection. In case that better detection rate are desired other than better false alarm rate, we should take a larger value for λ_1 as compared to a smaller value for λ_2; and vice versa.

5.2.3. Estimation of interference

From the intrusion detection algorithm, it is obvious that the accuracy of the values t_R and E[I_{max}] may significantly affect the intrusion detection rate and false alarm rate. To estimate the maximum interference, we use one-dimensional *Kalman* filtering technique. The technique not only guarantees a high level of prediction accuracy by filtering out measurement errors as well as by preserving the quick changes in interference power, but also achieves low computation overhead.

The operation assumptions for the wireless networks under consideration are as follows:

- It is assumed that a radio channel in a TMDA network is used, where time is divided into *slots* or *sessions*. These slots or sessions are reserved under

medium access control (MAC) protocol for a specific flow. During the reserved slots or sessions, only the transmitting node of the flow is allowed to send data onto the given channel and all nodes should keep silent. Multiple contiguous time slots or sessions can be used by the same transmitter for sending a data burst.

- Interference power in each time slot or session can be easily calculated, but with errors at each receiver. The interference power refers to the difference between the total received power and the power of the signal sent within the reserved slots or sessions, which is calculated based on the throughput and the total received power using Shannon-Hartley theory.

1) Kalman filters

Kalman filter is a *recursive data processing algorithm* with the purpose of estimating the state of a system from measurements which contain random errors [69]. The filter processes all available measurements in estimation of the current value of the variables, regardless of the precision of the measurements. It uses knowledge of the following aspects:

- Knowledge of the system and measurement device dynamics;
- The statistical description of the system noises, measurement errors, and uncertainty in the dynamics models;
- Any available information about initial conditions of the variables of interest.

There are three basic assumptions in the Kalman filter formulation:

- The system can be described with a *linear* model. Suppose the values of the variable we want to estimate at certain times t_0, t_1, t_2, etc are $\mathbf{x}_0$, $\mathbf{x}_1$, $\mathbf{x}_2$, etc. Then the value $\mathbf{x}_{k+1}$ at time t_{k+1} can be represented with a linear dynamic equation: $\mathbf{x}_{k+1} = \mathbf{\Phi x}_k + \mathbf{w}_k$.

- The system and measurement noises are *white* and *Gaussian*. That is, the noise is not correlated in time and the mean of the noise is zero.

The measurement value of $\mathbf{x}_k$ can be denoted as $\mathbf{z}_k = \mathbf{\Phi x}_k + \mathbf{v}_k$ (where $\mathbf{v}_k$ denotes the measurement noise). Under these three assumptions, the Kalman filter can be shown to be the best filter of any conceivable form.

By the Kalman filter theory [69], the time update equations are:

$$\tilde{\mathbf{x}}_{k+1} = \hat{\mathbf{x}}_k \tag{2}$$

$$\tilde{\mathbf{P}}_{k+1} = \hat{\mathbf{P}}_k + \mathbf{Q}_k \tag{3}$$

And the measurement update equations are:

$$\mathbf{K}_k = \tilde{\mathbf{P}}_k [\tilde{\mathbf{P}}_k + \mathbf{R}_k]^{-1} \tag{4}$$

$$\hat{\mathbf{x}}_k = \tilde{\mathbf{x}}_k + \mathbf{K}_k [\mathbf{z}_k - \mathbf{x}_k] \tag{5}$$

$$\hat{\mathbf{P}}_k = [1 - \mathbf{K}_k] \tilde{\mathbf{P}}_k \tag{6}$$

where $\tilde{\mathbf{x}}_k$ and $\hat{\mathbf{x}}_k$ are the *a priori* and *a posteriori* estimates of $\mathbf{x}_k$ respectively, $\tilde{\mathbf{P}}_k$ and $\hat{\mathbf{P}}_k$ are the *a priori* and *a posteriori* estimate-error variances, $\mathbf{K}_k$ is the Kalman gain, and $\mathbf{Q}_k$ and $\mathbf{R}_k$ are the covariance matrices for the process noise $\mathbf{w}_k$ and measurement noise $\mathbf{v}_k$ respectively.

For more details about Kalman filter, see [69], [70] and [71].

2) Estimation of signal interference using Kalman filter

By the Kalman filter theory, it is assumed in our estimation that:

- The signal interferences at certain times when we want to estimate can be described with a linear system, such as: $x_{k+1} = \phi x_k + \omega_k$. ω_k is the process noise at time t_k.

- The process and measurement noises on signal interferences are *white* and *Gaussian*.

Actually in this detection scheme, the signal interference is not measured directly but calculated based on the measurement of the throughput and the total received power during the reserved time slots or sessions. According to the Shannon-Hartley theorem, the calculated signal interference at time t_k is:

$$x_k = \frac{R_k}{2^{C_k / B}}$$

where R_k is the total received power at time t_k, while C_k is the throughput or capacity of the channel on the flow at time t_k. Please note that, using the Shannon-Hartley theorem, we are assuming that this C_k (or capacity) is actually the theoretical maximum capacity. Therefore the calculated interference, x_k, should be smaller than the real interference strength, and this may affect the detection rate of our intrusion detection mechanism.

We assume that the measurement on the throughput is accurate (this assumption is reasonable because the throughput can be easily measured at the receiving node) and that the measurement noises on the total received power are *white* and *Gaussian*, it can be proved that the measurement noises on the signal interference are *white* and *Gaussian* as well.

The process of applying Kalman filter on estimate of signal interferences is illustrated in Figure 23.

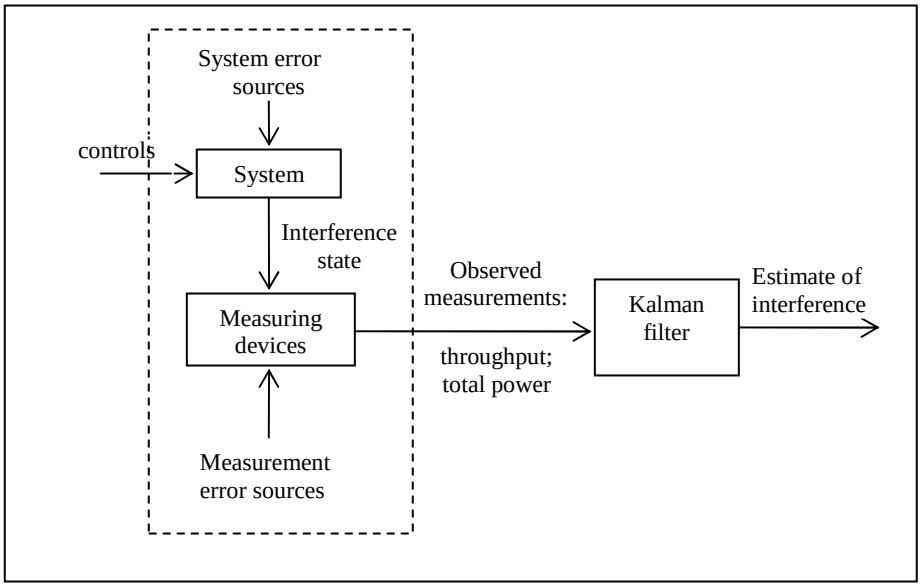


Figure 23. Application of Kalman filter on estimate of signal interferences

We use one-dimensional Kalman filter to estimate the signal interference at time t_k. According to the theory, the estimation algorithm is as follows:

Initialization: $P = \sigma^2; R = \pi^2$

For *t* = 0 : dt : duration
Begin

Take input of the measurement*:*

R = received power; *C = throughput/dt*;

$z = \frac{R}{2^{C/B}}$ %interference measurement

Recalculate *P* using (7);

$Inn = z - xhat;$ %innovation

$S = P + R;$ %covariance of innovation

$k = P * 1/S;$ %Kalman filter

$xhat = xhat + k * Inn;$ % a posteriori estimate of interference

$P = [1-k]P = (1-k)P;$ %covariance of prediction error

End.

Figure 24. Estimate algorithm of signal interferences

In Figure 24, $P\,(=p^2)$ and $R\,(=r^2)$ are the respective variances for the changes of interference power (or process noises) and the interference measurement error. In our initialization, we obtain the initial value of process noise σ by using measurement in a sliding window of *W* slots. We obtain the average changes of interference power from one time slot to the next by

$$\bar{\sigma}_k = \frac{1}{W}\sum_{l=k-W+1}^{k} \sigma_l - \sigma_{l-1} \qquad (7)$$

Then the approximated variance of process noise is

$$P = \sigma^2 = \frac{1}{W-1}\sum_{l=k-W+1}^{k} [(\sigma_l - \sigma_{l-1}) - \bar{\sigma}_k]^2 \qquad (8)$$

Note that σ_I includes the interference measurement errors *r*, which have a Gaussian distribution with zero mean. Therefore, *W* should be large enough (e.g. $W \geq 1000$) to give an unbiased estimate of average changes of interference power in consecutive time slots.

The variance of the interference measurement error *R* depends on the noise level and the error characteristics of the measurement circuit in use. In practice, the initial value of $R(=\pi^2)$ can be determined by measuring the "received" power on an idle channel with known thermal noises. Therefore, the variance of the "received" power over a time window can serve as an estimate of π^2.

3) Some discussions on the estimation

There is such possibility that the measurement noise power level is not constant or the noise is actually time correlated. But in these instances, a white noise put through a small linear system can duplicate virtually any form of time-correlated noise. This system, called a "shaping filter," can then be added to the original system to achieve an overall linear system driven by white noise once again.

In a real network, the signal interferences may not only come from far transmissions, but also from the thermal noises at the background. This is one of the main causes for the measurement noises.

Some researchers proposed two-dimensional Kalman filter to estimate signal interferences. In the approach of estimation with two-dimensional filter, the number of co-channel interferers is also considered to enhance the accuracy of interference power prediction. We argue that two-dimensional filter may improve the accuracy of estimation in some circumstances, but it certainly introduces much more computational overhead than the one-dimensional filter technique that is proposed in this chapter. Moreover, the

number of interferers is very difficult to predict at any time in that it is correlated to the node mobility pattern in the vicinity as well as their traffic patterns, which is the information that is hard for a node to obtain.

5.3. Simulation and Performance Analysis

We can see from the design of the detection mechanism that use of the Shannon-Hartley theorem (Equation (1)) and the interference estimation approach using Kalman filter has significant impact on the performance of the detection mechanism. Therefore, we first do some simulation to evaluate the errors in using Shannon-Hartley theorem and the interference estimation. Then we conduct performance analysis of our detection mechanism based on simulation. We still use ns-2 to investigate the performance of the proposed approaches.

Our simulation is based on a 1500 by 300 meters rectangle space. 50 nodes move from a random starting position to a random destination with a random speed uniformly distributed between 0 to 20 m/sec. The pause time is set to 600 seconds.

The channel capacity of mobile hosts is set to the same value: 2 Mbps. We assume all nodes have the same transmission range of 250 meters at the beginning of the simulation. Their transmission ranges afterwards depend on their remaining battery level. *Two-ray ground reflection model* is used as the channel model.

5.3.1. Performance analysis on use of Shannon-Hartley theorem and evaluation on interference estimation

We evaluate our interference estimation algorithm by simulation. To evaluate the algorithm separately from the Shannon-Hartley theorem, we remove the use of Shannon-Hartley in the algorithm. Instead, we use the real interference value in the algorithm to replace the “interference measurement” with Shannon-Hartley theorem.

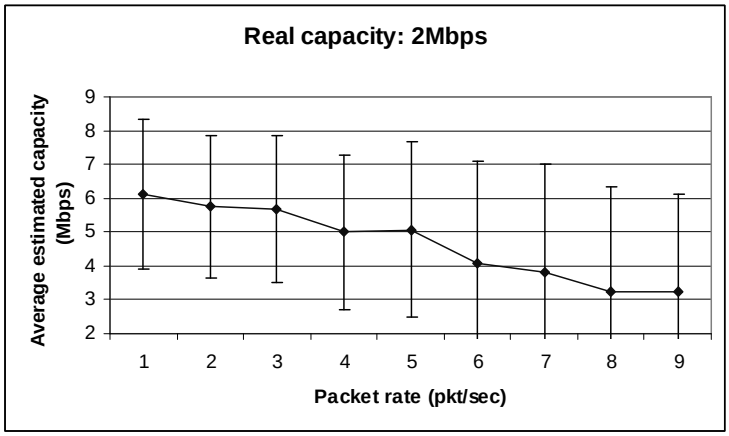


Figure 25. Average estimated capacity (real capacity: 2Mbps)

In our simulation, we set the real bandwidth as 2Mbps. The simulation results are shown in Figure 25. We found that with increase of the packet rate, the estimated capacity is closer to the real capacity value. That is, the accuracy increases when the data packet rate increases. This may be caused by decreased signal-to-noise value in the theorem.

We calculate the accuracy of the interference estimation as $1-\frac{|estimated-real|}{real}$, where "*estimated*" denotes "estimated interference value" and "*real*" denotes "real interference value". The simulation results are demonstrated in Figure 26.

We can see from Figure 26 that the accuracy of the estimation algorithm decreases with increase of the data packet rate. This may result from the fact that the deviation of the interference or noise strength increases when the traffic load increases in the network.

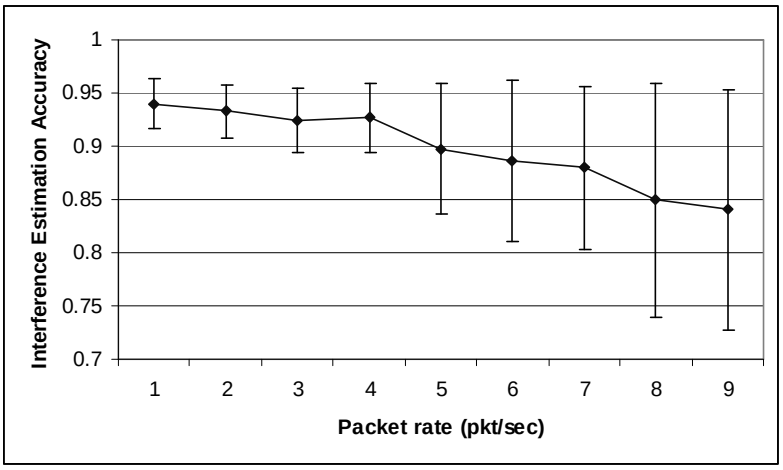


Figure 26. Accuracy of the interference estimation algorithm

5.3.2. Performance evaluation on the intrusion detection mechanism

We simulate DoQoS and QoS attacks in separate scenarios. For any given scenario, we run the simulation for 40 minutes to get the data. For the first 250 seconds, each node collects the data of interference power every 250ms during 1000 time slots (W=1000) to calculate the initial value of process noise by feeding the data into formulae (6) and (7). Then we use the remaining time to simulate the attacks and to test our approaches.

At the beginning of the simulation, we randomly pick a node as the malicious node, who continuously sends packets regardless of legal traffic from other nodes. During the simulation, we also randomly select a node in the attacker's neighborhood as the receiving node, whose receiving will be affected by the attacker's malicious behaviors. Then the receiving node will conduct the detection. When the attacker or the receiving node moves out of the neighborhood, new receiving node will be randomly selected. The packet rate of the malicious node is always the same as the packet rate of the legal flow.

Because thermal noises are not simulated in the simulator (i.e. $t_R = 0$), we use the value of the adjustment factor for QoS attack detection λ_1 as the QoS attack detection threshold, and set λ_1 as the minimum signal interference that has been collected during the first 250 seconds from calculation of the initial process noise:

$$\lambda_1 = \min_{j=1,2,\ldots,1000}(\sigma_j)$$

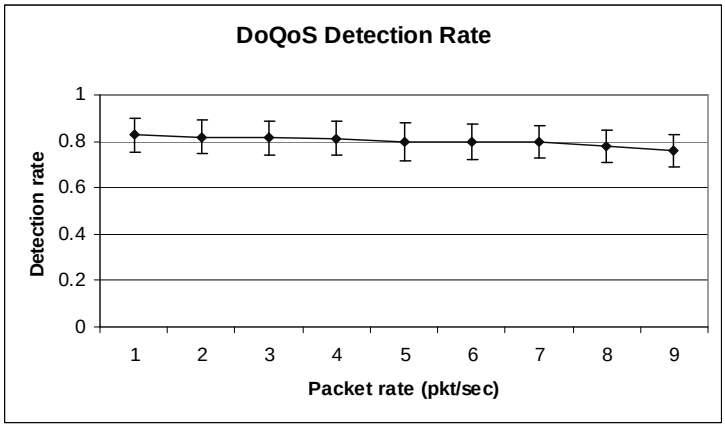
We set the second security factor as zero, i.e. $\lambda_2 = 0$.

We use the following two metrics to evaluate the performance of the intrusion detection mechanism:

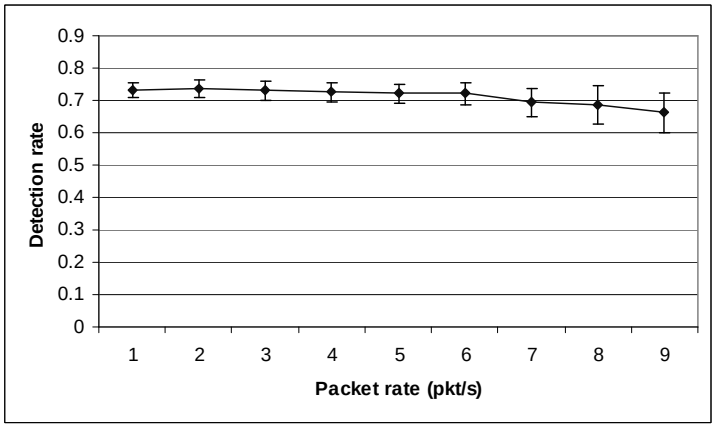
Detection rate: It is defined as the proportion of the number correct alarms of malicious attacks to the total number of alarms that should be reported.

Figure 27 demonstrated the detection rates for DoQoS and QoS attacks respectively.

From Figure 25, Figure 26 and Figure 27 (a), we observe that the detection rate of DoQoS is significantly affected by the interference estimation algorithm. The detection rate does not fit the trend of the accuracy in calculation with Shannon-Hartley theorem. The reason may be that the interference is estimated based on the values that are also calculated with the Shannon-Hartley theorem and therefore the accuracy of the calculation is not reflected in the detection.



(a) Detection rate for DoQoS attacks



(b) Detection rate for QoS attacks

Figure 27. Detection rate for DoQoS and QoS attacks

False positive rate: It is defined as the percentage of decisions in which benign behaviors are flagged as anomalous. We evaluate this metrics by simulating the environment where there does not exist any malicious node.

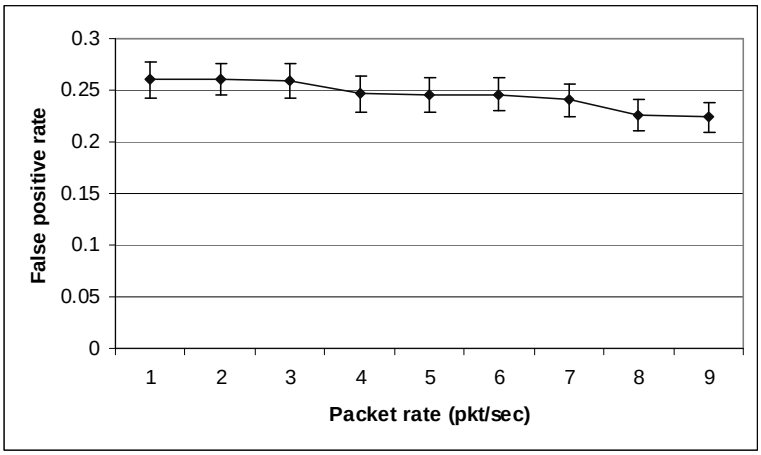


Figure 28. False positive rate for DoQoS attack detection

Figure 28 demonstrates the false positive rates for DoQoS attack detection. The false positive rates for QoS attack detection are always below 0.22%.

5.4. Conclusion

MANETs are vulnerable to QoS and DoQoS attacks due to the characteristics of open shared medium and network topology change. In this chapter, we propose a security mechanism to detect intrusions on bandwidth reservation in MANETs.

6. CONCLUSION AND FUTURE WORK

MANETs are characterized by the absence of fixed infrastructure, rapid topology change and high node mobility. These characteristics impose tremendous difficulty on design and implementation of security mechanisms that provide security protection or intrusion detection.

Security is a critical issue and offers serious challenges in QoS provisioning in wireless ad hoc networks. Without protection from security mechanisms, a QoS system is vulnerable to various malicious attacks. Yet there is little work published in this area up to date.

This research has filled in this blank via providing security mechanisms that can prevent MANET QoS mechanisms from being tampered by malicious adversaries. The approaches such as intrusion prevention and intrusion detection have been applied to guarantee an advanced security level.

6.1. Contributions

In this dissertation, we have addressed security issues for QoS systems in Mobile Ad Hoc Networks. We have designed a secure QoS system to prevent from or detect various malicious attacks from different aspects. The security mechanisms we designed can be utilized to preserve protected information and network resources, therefore can protect QoS from being tampered with by adversaries.

The major contributions of this research include:

6.1.1. A lightweight authentication protocol for MANETs

We have proposed a lightweight authentication protocol that can effectively and efficiently provide security properties such as authenticity and integrity for communicating neighbor nodes in MANETs. The protocol utilizes one-way hash chains to compute authentication keys, which not only eliminates the high performance overhead imposed by asymmetric cryptography (such as digital signatures), but also avoids the difficulty of key management introduced by secret paired symmetric key. The protocol also used delayed key disclosure to prevent a malicious entity from forging packets with Message Authentication Codes (MACs) with an already released key. The authentication protocol is lightweight, scalable and tolerant of packet loss. The performance analysis showed that the protocol incurs low overhead penalty and also achieves a tradeoff between security and performance.

6.1.2. Security in QoS models and signaling systems for MANETs

In this dissertation, we analyzed the vulnerabilities and types of security violations for MANET QoS models, which include IntServ model, Diffserv model and the Flexible QoS Model for MANETs (FQMM). The analysis demonstrated that DiffServ and FQMM are vulnerable to attacks such as theft and depletion of network resources. Compared to the DiffServ model, the IntServ approach does not have the security risks mentioned above because it is based on flows rather than on aggregated traffic as in DiffServ and FQMM. However, IntServ model requires a signaling system to achieve QoS provision along a data path. Without protection of certain security mechanisms, a QoS signaling system can still become the target of malicious attacks.

In order to detect and prevent from misbehaviors on QoS signaling systems, we have proposed a Secure Mechanism for QoS Signaling system in MANETs. In this dissertation, we have proposed a security mechanism for MANET QoS signaling systems. The mechanism is able to efficiently detect intrusions on QoS parameters

transmitted over a path in the absence of adjacent colluding nodes. The simulation results have demonstrated that the proposed system achieved good performance with fairly high detection rate and low delay penalty.

6.1.3. Intrusion detection for bandwidth reservation in MANETs

In the traditional Internet, if the violation on bandwidth reservation exceeds a predefined threshold, we can conclude that an intrusion has happened. In MANETs, however, due to the characteristic of high node mobility and dramatic capacity change on communication links, a node can only promise not to deliberately oversubscribe itself and not to intentionally prevent the resources from being available. QoS cannot be guaranteed and a break of QoS promise can result from malicious attacks as well as radio interference from the nodes who just “wandered” into the neighborhood unaware of the reservation. Moreover, communication links in MANETs are open medium and therefore subject to radio interference. Detection of intrusion on bandwidth reservation needs to distinguish these cases and apparently is not a trivial task.

We designed an algorithm to detect both DoS attacks (issued by malicious nodes in the neighborhood to disrupt the service), and QoS attacks (issued by relay node on the path to disrupt the service or to steal the bandwidth). Simulation and performance evaluation of the algorithm are also demonstrated.

6.2. Future Work

At this point, we have focused on intrusion detection for bandwidth reservation. Besides bandwidth, an adversary can also target other Quality of Service parameters (such as *delay* and *jitter*), which will also cause violation of reserved QoS. In the future, we will design security techniques to thwart this type of attacks. The technique to detect service violation on delay or jitter may require an upstream node selectively promiscuously

listen to the downstream nodes and monitor whether the downstream node is providing promised QoS.

In the design of the lightweight authentication protocol, we used *delayed key disclosure* to prevent malicious entities from forging packets with Message Authentication Codes using an already released key. The impact of the delayed key disclosure on the authentication will be analyzed in the future. In addition, an algorithm to determine the value of key disclosure delay is also worth further investigation.

Our future research direction also includes the implementation of the secure QoS system that we have proposed in a real mobile ad hoc network and to evaluate its performance in the real world.

REFERENCES

[1] R. Braden, D. Clark, and S. Shenker, "Integrated Services in the Internet Architecture: an Overview," in *RFC 1633*, June 1994, http://www.ietf.org/rfc/rfc1633.txt.

[2] S. Blake, D. Black, and M. Carlson, "An Architecture for Differentiated Service," in *RFC 2475*, December 1998, http://www.ietf.org/rfc/rfc2475.txt.

[3] H. Xiao, W.K.G. Seahand, A. Lo, K.C. Chua, "A Flexible Quality of Service Model for Mobile Ad-Hoc Networks," in *Proc. of IEEE Vehicular Technology Conf. (VTC2000)*, Tokyo, Japan, May 2000, pp. 445-449.

[4] D. Maltz, "Resource Management in Multi-hop Ad Hoc Networks," Technical report, *CMU-CS-00-150*, Carnegie Mellon University, School of Computer Science: Pittsburgh, PA, November 1999.

[5] F. Stajano, and R. Anderson, "The Resurrecting Duckling: Security Issues for Ad-hoc Wireless Networks," in *Proc. 7th International Workshop on Security Protocols*, Cambridge, UK, April 1999, pp. 172-194.

[6] K. Nahrstedt, "Can Network QoS and Security Live in Symbiosis?" in *Proc. of National Information Systems Security Conf. (NISSC)*, Baltimore, MD, October 2000, pp. 584-585.

[7] F. Baker, B. Lindell, and M. Talwar, "RSVP Cryptographic Authentication," in *RFC 2747*, January 2001, http://www.ietf.org/rfc/rfc2747.txt.

[8] A. Talukdar, B.R. Badrinath, and A. Acharya, "MRSVP: A Resource Reservation Protocol for an Integrated Services Network with Mobile Hosts," *Wireless Networks*, vol. 7, no. 1, pp. 5-19, 2001.

[9] C. Tseng, G. Lee, and R. Liu, "HMRSVP: A Hierarchical Mobile RSVP Protocol," in *Proc. of 2001 International Conf. on Distributed Computing Systems Workshop*, Mesa, AZ, April 2001, pp. 467-472.

[10] S.B. Lee and A.T. Campbell, "INSIGNIA: In-band Signaling Support for QOS in Mobile Ad Hoc Networks," in *Proc. of 5th International Workshop on Mobile Multimedia Communications (MoMuC'98)*, Berlin, Germany, October 1998, pp. 445-449.

[11] C.E. Perkins and E.M. Belding-Royer, "Quality of Service for Ad Hoc On-Demand Distance Vector Routing," in *Internet Draft*, November 2001, http://www.ietf.org/internet-drafts/draft-perkins-manet-aodvqos-01.

[12] T. Chen, "Efficient Routing and Quality of Service Support for Ad Hoc Wireless Networks," Ph.D. Dissertation, Computer Science Department, University of California at Los Angeles, Los Angeles, CA, 1998.

[13] C. Lin and J. Liu, "QoS Routing in Ad Hoc Wireless Networks," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 8, pp. 1426-1438, August 1999.

[14] S. Chen and K. Nahrstedt, "Distributed Quality-of-Service Routing in the Ad-Hoc Networks," *IEEE Journal on Special Areas in Communications*, vol. 17, no. 8, pp. 1488-1505, October 1998.

[15] Q. Xue and A. Ganz, "Ad Hoc QoS On-demand Routing (AQOR) in Mobile Ad Hoc Networks," *Journal of Parallel and Distributed Computing*, vol. 63, no. 2, pp. 154-165, February 2003.

[16] R. Sivalumar, P. Sinha, and V. Bharghavan, "CEDAR: a Core-Extraction Distributed Ad Hoc Routing Algorithm," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 8, pp. 1454-1465, August 1999.

[17] Y. Zhang and W. Lee, "Intrusion Detection in Wireless Ad-Hoc Networks," in *Proc. Sixth Annual International Conf. on Mobile Computing and Networking* Boston, MA, August 2000, pp. 275-283.

[18] C.E. Perkins and P. Bhagwat, "Routing over Multihop Wireless Network of Mobile Computers," *SIGCOMM '94: Computer Communications Review*, vol. 24, no. 4, pp. 234-244, October 1994.

[19] P. Jacquet and T. Clauseen, "Optimized Link State Routing Protocol," in *RFC 3626*, October 2003, http://www.ietf.org/rfc/rfc3626.txt

[20] C.E. Perkins, E.M. Royer, and S.R. Das, "Ad Hoc on-Demand Distance Vector (AODV) Routing," in *Internet Draft*, June 2002.

[21] D.B. Johnson, D. Maltz and Y. Hu, "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks," in *Internet Draft*, April 2003, http://www.ietf.org/internet-drafts/draft-ietf-manet-dsr-09.txt

[22] V. Park and S. Corson, "Temporally-Ordered Routing Algorithm (TORA) Version 1 - Functional Specification," in *Internet Draft*, November 1997, http://www.ietf.org/proceedings/02mar/I-D/draft-ietf-manet-tora-spec-04.txt

[23] Z.J. Haas and M.R. Pearlman, "The Zone Routing Protocol (ZRP) for Ad Hoc Networks," in *Internet Draft*, July 2002, http://www.ietf.org/proceedings/02nov/I-D/draft-ietf-manet-zone-zrp-04.txt

[24] R. Ramanujan, S. Kudige and T. Nguyen, "Techniques for Intrusion-Resistant Ad Hoc Routing Algorithms (TIARA)," in *Proc. of 21st Century Military Communications Conf. (MILCOM 2000)*, Los Angeles, CA, October 2000, pp. 660-664.

[25] Y. Hu, A. Perrig, and D.B. Johnson, "Ariadne: A Secure On-Demand Routing Protocol for Ad Hoc Networks," in *Proc. International Conf. on Mobile Computing and Networking*, Atlanta, GA, 2002, pp. 12-23.

[26] K. Sanzgiri, B. Dahill, B.N. Levine, C. Shields, and E.M. Belding-Royer, "A secure Routing Protocol for Ad Hoc Networks," in Technical Report: UM-CS-2001-037, University of Massachusetts, Amherst, MA, August 2001.

[27] M.G. Zapata, "Secure Ad Hoc On-Demand Distance Vector (SAODV) Routing," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 6, no. 3, pp. 106-107, July 2002.

[28] P. Papadimitratos and Z.J. Haas, "Secure Routing for Mobile Ad Hoc Networks," in *Proc. SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS 2002)*, San Antonio, TX, January 2002, pp. 27-31.

[29] Y. Hu, D.B. Johnson, and A. Perrig, "SEAD: Secure Efficient Distance Vector Routing for Mobile Wireless Ad Hoc Networks," in *Proc. Fourth IEEE Workshop on Mobile Computing Systems and Applications (WMCSA '02)*, New York, NY, June 2002, pp. 3-13.

[30] S. Yi, P. Naldurg, and R. Kravets, "Security-Aware Ad Hoc Routing for Wireless Networks," in *Proc. International Conf. on Mobile Computing and Networking (MobiHoc 2001)*, Long Beach, CA, October 2001, pp. 299-302.

[31] H. Yang, X. Meng, and S. Lu, "Self-Organized Network-Layer Security in Mobile Ad Hoc Networks," in *Proc. International Conf. on Mobile Computing and Networking*, Atlanta, GA, 2002, pp. 11-20.

[32] Y. Hu, A. Perrig, and D.B. Johnson, "Packet Leashes: a Defense Against Wormhole Attacks in Wireless Networks," in *Proc. Twenty-Second Annual Joint Conf. of the IEEE Computer and Communications Societies (INFOCOM 2003)*, San Francisco, CA, April 2003, pp. 1976-1986.

[33] C. Castelluccia and G. Montenegro, "Protecting AODV Against Impersonation Attacks," *ACM Mobile Computing and Communications Review* (*SIGMOBILE*), vol. 6, no. 3, pp. 108-109, July 2002.

[34] P. Brutch and C. Ko, "Challenges in Intrusion Detection for Wireless Ad-Hoc Networks," in *Proc. of Symposium on Applications and the Internet Workshops*, Orlando, FL, January 2003, pp. 368-373.

[35] J. Kong, H. Luo, K. Xu, D. Gu, and M. Gerla, "Adaptive Security for Multi-layer Ad-hoc Networks," Special Issue of *Wireless Communication and Mobile Computing*, vol. 2, no. 5, pp. 533-547, 2002.

[36] L. Zhou, and Z.J. Haas, "Securing Ad Hoc Networks," *IEEE Networks*, vol. 13, no. 6, pp. 24-30, December 1999.

[37] S. Capkun, L. Buttyan, and J.P. Hubaux, "Self-Organized Public-Key Management in Ad Hoc Wireless Networks," in *Proc. of ACM Workshop on Wireless Security (WiSe)*, Atlanta, GA, September 28, 2002, pp. 52-64.

[38] X. Li, Y. Wang, and O. Frieder, "Efficient Hybrid Key Agreement Protocol for Wireless Ad Hoc Networks," in *Proc. IEEE 11th International Conf. on*

Computer Communications and Networks (ICCCN2002), Miami, FL, October 2002, pp. 404 – 409.

[39] K. Weniger, "Passive Duplicate Address Detection in Mobile Ad Hoc Networks," in *Proc. Wireless Communications and Networking, 2003 (WCNC 2003*), New Orleans, LA, March 2003, pp. 1504-1509.

[40] G. O'Shea and M. Roe, "Child-Proof Authentication for MIPv6 (CAM)," *ACM SIGCOMM Computer Communication Review*, vol. 31, no. 2, pp. 4-8, April 2001.

[41] G. Montenegro and C. Castelluccia, "Statistically Unique and Cryptographically Verifiable (SUCV) Identifiers and Addresses," in *Network and Distributed System Security Symposium*, San Diego, CA, February 2002, pp. 65-77.

[42] V. Talwar and K. Nahrstedt, "Securing RSVP for Multimedia Applications," in *Proc. ACM Multimedia Security Workshop*, New York, NY, November 2000, pp. 153-156.

[43] R. Braden and L. Zhang, "RSVP Cryptographic Authentication -- Updated Message Type Value," in *RFC 3097*, April 2001, http://www.ietf.org/rfc/rfc3097.txt

[44] T. Wu, S. Wu, F. Gong, Z. Fu, and H. Huang, "Securing QoS: threats to RSVP messages and their countermeasures," in *Proc. Seventh International Workshop on Quality of Service*, London, UK, June 1999, pp. 62-64.

[45] Z. Fu, "Network Management and Intrusion Detection for Quality Of Network Services," Ph.D. Dissertation, in Computer Science Department, North Carolina State University, Raleigh, June 2001.

[46] V. Talwar, S.K. Nath, and K. Nahrstedt, "RSVP-SQoS: A Secure RSVP Protocol," in *Proc. International Conf. on Multimedia and Exposition*, Tokyo, Japan, August 2001, pp. 579-582.

[47] E. Fulp, Z. Fu, D.S. Reeves, S. Wu and X. Zhang "Preventing denial of service attacks on quality of service," in *Proc. DARPA Information Survivability Conf. & Exposition II*, Anaheim, CA, June 2001, pp. 159-172.

[48] S. Chen, "Routing Support for Providing Guaranteed End-to-End Quality of Service," Ph. D. Dissertation, in Computer Science Department, University of Illinois at Urbana-Champaign, Urbana-Champaign, IL, 1999.

[49] P. Sethi and G. Barua, "CRESQ: Providing QoS and Security in Ad Hoc Networks," in *Proc. Eleventh Euromicro Conf. on Parallel, Distributed and Network-Based Processing*, Genova, Italy, February 2003, pp. 544-550.

[50] S. Yi, P. Naldurg, and R. Kravets, "Integrating Quality of Protection into Ad Hoc Routing Protocols," in *Proc. 6th World Multi-Conference on Systemics, Cybernetics and Informatics*, Orlando, FL, July 2002, pp. 551-557.

[51] A. Perrig, R. Canetti, J. Tygar, and D. Song, "Efficient Authentication and Signing of Multicast Streams over Lossy Channels," in *Proc. of IEEE Symposium on Security and Privacy*, Berkeley, CA, May 2000, pp. 56-73.

[52] S. Zhu, S. Xu, S. Setia, and S. Jajodia, "LHAP: A Lightweight Hop-by-Hop Authentication Protocol for Ad-Hoc Networks," in *Proc. 23rd International Conf. on Distributed Computing Systems Workshops (ICDCSW 2003)*, Providence, RI, May 2003, pp. 749-755.

[53] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J. D. Tygar, "SPINS: Security Protocols for Sensor Networks," in *Proc. of the 7th Annual International Conference on Mobile Computing and Networking (MobiCom '01)*, Rome, Italy, July 2001, pp. 189-199.

[54] R. Rivest, "The MD5 Message-Digest Algorithm", *RFC 1321*, April 1992, http://www.faqs.org/rfcs/rfc1321.html.

[55] NIST, FIPS PUB 180-1: "Secure Hash Standard", April 1995, http://www.itl.nist.gov/fipspubs/fip180-1.htm.

[56] H. Dobbertin, "The Status of MD5: After a Recent Attack", RSA Labs' *CryptoBytes*, vol. 2, no. 2, 1996, pp. 1-6.

[57] H. Krawczyk, M. Bellare and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication," *RFC 2104*, February 1997, http://www.faqs.org/rfcs/rfc2104.html.

[58] "The network simulator - ns-2," February 2005, http://www.isi.edu/nsnam/ns/.

[59] M. Brown, D. Cheung, D. Hankerson, J. Hernandez, M. Kirkup and A. Menezes "PGP in Constrained Wireless Devices," in *Proc. 9th USENIX Security Symposium*, Denver, CO, August 2000, pp. 247-261.

[60] S. Marti, T. Giuli, K. Lai and M. Baker, "Mitigating Routing Misbehavior in Mobile Ad Hoc Networks," in *Proc. of the Sixth Annual International Conf. on Mobile Computing and Networking (MobiCom '00)*, Boston, MA, August 2000, pp. 255-265.

[61] S. Zhong, Y. R. Yang and J. Chen, "Sprite: A Simple, Cheat-Proof, Credit-Based System for Mobile Ad Hoc Networks," in *Proc. of IEEE Conf. on Computer Communications (INFOCOM'03)*, San Francisco, CA, March 2003, pp. 1987-1997.

[62] U. D. Black, *Internet Security Protocols: Protecting IP Traffic*, Upper Saddle River, NJ: Prentice Hall PTR, 2000.

[63] B. Bhargava, "Detecting Service Violations in Internet and Mobile Ad Hoc Networks," talk at Purdue University, http://www.cs.purdue.edu/homes/bb/nwu-ece.ppt.

[64] C. Zhu and M.S. Corson, "QoS Routing for Mobile Ad Hoc Networks"**,** in *Proc. 21st Annual Joint Conf. of the IEEE Computer and Communications Societies, (INFOCOM 2002)*, New York, NY, June 2002, pp. 958-967.

[65] L. Georgiadis, P. Jacquet and B. Mans, "Bandwidth Reservation in Multihop Wireless Networks: Complexity and Mechanisms," in *Proc. 24th International Conf. on Distributed Computing Systems Workshops*, Tokyo, Japan, March 2004, pp. 76-767.

[66] T. D. Huynh, N. R. Jennings and N. Shadbolt, "Fire: An Integrated Trust and Reputation Model for Open Multi-Agent Systems," in *Proc. 16th European Conference on Artificial Intelligence*, Valencia, Spain, August 2004, pp. 18-22.

[67] P. Michiardi and R. Molva, "Core: A Collaborative Reputation Mechanism to Enforce Node Cooperation in Mobile Ad Hoc Networks," in *Proc. of the IFIP TC6/TC11 Sixth Joint Working Conference on Communications and Multimedia Security: Advanced Communications and Multimedia Security*, Portoroz, Slovenia, 2002, pp. 107-121.

[68] S. Buchegger and J.-Y. L. Boudec, "Performance Analysis of the Confidant Protocol (Cooperation of Nodes: Fairness in Dynamic Ad-Hoc Networks)," in *Proc. of IEEE/ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHOC)*, Lausanne, Switzerland, June 2002, pp. 226-236.

[69] P. Maybeck, *Stochastic Models, Estimation, and Control*, New York: Academic Press, 1979.

[70] R.G. Brown and P.Y.C. Hwang, *Introduction to Random Signals and Applied Kalman Filtering*, Second Edition, New York: John Wiley & Sons, Inc, 1992.

[71] M. S. Grewal, and A. P. Andrews, *Kalman Filtering Theory and Practice*, Upper Saddle River, NJ: Prentice Hall, 1993.

APPENDIX A

The simulation model for the authentication protocol that we presented in Chapter III and the data we obtained in the simulation are described in this appendix.

The routing protocol we used in our simulation is AODV. The Medium Access Control (MAC) protocol is IEEE 802.11 and the Transportation layer protocol is User Datagram Protocol (UDP), which are both available as a part of the simulator. The size of data packets is 512 bytes the traffic sources are Constant-Bit-Rate (CBR). We assume all the nodes have the same initial transmission range of 250 meters.

In our simulation, all traffic is generated and the statistical data are collected after a warm-up time of 100 seconds in order to allow the network to finish initialization process.

The data for our simulation are shown in Table A-1 through A-6.

Table A-1 shows the data that we use to generate Figure 6, which presents the resent rate of the KEYUPDATE messages.

Table A-1. Data for "resent rate of KEYUPDATE messages"

Packet rate (pkt/sec)	Resent Percentage (%)	Deviation (%)
2	2.29619	0.40325
4	2.88991	0.4218
6	3.05035	0.71746
8	3.19447	0.72237
10	3.41463	0.75306

Table A-2 shows the data that we use to generate Figure 7 (a) , which presents the average hop-by-hop delay in the scenario of 9 nodes (Scenario 1 described in the chapter).

Table A-2. Data for "average hop-by-hop delay: scenario of 9 nodes"

Packet rate (pkt/sec)	Average hop-by-hop delay (sec)	deviation
2	0.006028	1.25E-06
6	0.006958	7.05E-06
4	0.007004	7.55E-06
8	0.007704	1.38E-05
10	0.008237	2.20E-05

Table A-3 shows the data that we use to generate Figure 7 (b), which presents the average hop-by-hop delay in the scenario of 50 nodes (Scenario 2 described in the chapter).

Table A-3. Data for "average hop-by-hop delay: scenario of 50 nodes"

Packet rate (pkt/sec)	Average hop-by-hop delay (sec)	deviation
2	0.028315	0.005089
6	0.18477	0.167415
4	0.32784	0.300454
8	0.364751	0.334491
10	0.373874	0.364128

Table A-4 shows the data that we use to generate Figure 8, which presents the percentage of packets arriving safely.

Table A-5 shows the data that we use to generate Figure 9 (a), which presents the average dropped packet rate, with a cache of 16 packets size.

Table A-6 shows the data that we use to generate Figure 9 (b), which presents the average dropped packet rate, with a cache of 32 packets size.

Table A-4. Data for "Percentage of packets arriving safely"

Packet rate / Key disclosure delay (sec)	2 pkt/sec	4 pkt/sec	6 pkt/sec	8 pkt/sec	10 pkt/sec
0.2	98.6113	73.4077	61.5724	55.4833	55.9702
0.4	99.5772	85.454	75.0691	68.2309	68.7154
0.6	99.8411	91.1112	82.4131	75.8788	76.2754
0.8	99.9129	93.9528	86.7979	81.6012	81.7163
1	99.9462	95.5906	89.8889	85.3947	85.5514
1.2	99.9744	96.6865	92.1106	88.3968	88.3224
1.4	99.9949	97.3764	93.7613	90.6632	90.4576
1.6	100	97.9003	94.9893	92.406	92.2446
1.8	100	98.3402	95.9775	93.7332	93.7527
2.0	100	98.6862	96.737	94.8168	94.8451
2.2	100	98.9062	97.4068	95.6151	95.6622
2.4	100	99.0601	97.8954	96.25	96.2804
2.6	100	99.1741	98.2395	96.7787	96.8417
2.8	100	99.2861	98.4975	97.2031	97.2449
3.0	100	99.4001	98.686	97.6188	97.6517

Table A-5. Data for "average dropped packet rate, cache size: 16 pkt"

Disclosure delay / Packet rate (pkt/sec)	3 sec (%)	2 sec (%)
2	37.8599	6.7898
4	53.2993	29.9489
6	56.6632	34.9948
8	58.1664	37.2497
10	59.4185	39.1278

Table A-6. Data for "average dropped packet rate, cache size: 32 pkt"

Disclosure delay / Packet rate (pkt/sec)	3 sec (%)	2 sec (%)
2	0	0
4	6.5985	0
6	13.3263	0
8	16.3329	0
10	18.8371	0

APPENDIX B

The data for the signaling security mechanism that we presented in Chapter IV are described in this appendix.

We use the same simulation model that is used in Chapter III and described in Appendix A. We added a delay field to the AODV model to simulate the part of the QoS AODV protocol that is related to the performance evaluation on our mechanism.

Table B-1 shows the data that we use to generate Figure 16, which presents the average RREQ hop-by-hop delay.

Table B-1. Data for "average RREQ hop-by-hop delay"

Number of connections	AODV with delay field (ms)	Deviation (ms)	Security QoS (ms)	Deviation (ms)
10	9.023	0.98442	9.981	0.87649
20	10.001	0.98736	11.639	0.93208
30	11.475	0.98609	12.719	0.98214

Table B-2 shows the data that we use to generate Figure 17 (a), with 50 nodes, 20 maximum connections and packet rate of 4 pkt/sec.

Table B-2. Data for "Intrusion detection rate for QoS Signaling system, 50-20-4"

Pause time (sec)	Basic scheme	deviation	Enhanced scheme	deviation
200	0.764	0.054	0.933	0.004
400	0.837	0.049	0.975	0.001
600	0.841	0.049	0.985	0.001

Table B-3 shows the data that we use to generate Figure 17 (b), with 50 nodes, 20 maximum connections and 600 seconds of pause time.

Table B-3. Data for "Intrusion detection rate for QoS Signaling system, 50-600-20"

Packet rate (pkt/sec)	Basic scheme	deviation	Enhanced scheme	deviation
4	0.841	0.032	0.985	0.014
8	0.804	0.0318	0.945	0.026
10	0.781	0.0455	0.944	0.011

APPENDIX C

The simulation model and parameters we used in Chapter V is summarized in Table C-1.

Table C-1. Simulation setup and parameters for bandwidth reservation intrusion detection

Area	1500x300(m^2)
Propagation model	Two-ray ground reflection model
MAC protocol	IEEE802.11 with modification
Routing protocol	AODV
Initial Transmission range	250m
Node max speed	20m/s
Node pause time	60sec
Traffic type	UDP
Estimation initialization	W=1000, dt = 250ms
Security factors	$\lambda_1 = \min_{j=1,2,...,1000}(\sigma_j)$, $\lambda_2 = 0$
Misbehavior (DoQoS)	sends packets regardless of reservation
Misbehavior (QoS)	Leaves the bandwidth unused

The IEEE 802.11 protocol was modified to simulate bandwidth reservation. We make all the neighbors of the transmitting node and receiving node to keep silent during the simulation time, except for the attacker.

The two-ray ground reflection model is used to predict the received power based on the transmitted power and the distance of two nodes. The model is implemented in the ns-2 simulator. The model uses Friss-space attenuation ($1/r^2$) at near distances and an approximation to Two ray Ground ($1/r^4$) at far distances. The approximation assumes reflection off a flat ground plane. In the model, a cross-over distance r_c is first calculated:

$$r_c = (4\pi h_t h_r) / \lambda$$

where h_t and h_r are the heights of the transmission and receive antennas respectively, and λ is the wavelength.

Then if $r < r_c$, the received power is:

$$\Pr(r) = \frac{P_t G_t G_r \lambda^2}{(4\pi)^2 d^2 L}$$

If $r < r_c$, the received power at distance r is:

$$P_r(r) = \frac{P_t G_t G_r h_t^2 h_r^2}{r^4 L}$$

where P_t is the transmitted signal power, G_t and G_r are the antenna gains of the transmitter and the receiver respectively. L ($L \geq 1$) is the system loss. We use all the default values in ns-2: $G_t = G_r = 1$, $h_t = h_r = 1.5$ (m), L =1, $\lambda = 0.32823$.

Table C-2 shows the data that we use to generate Figure 25, which presents the average estimated capacity with Shannon-Hartley theorem. The real capacity is 2Mbps.

Table C-2. Data for "average estimated capacity using Shannon-Hartley theorem"

Packet rate (pkt/sec)	Estimated capacity(Mbps)	deviation
1	6.102	2.215
2	5.754	2.109
3	5.672	2.183
4	5.003	2.276
5	5.075	2.602
6	4.088	3.022
7	3.826	3.199
8	3.251	3.074
9	3.254	2.855

Table C-3 shows the data that we use to generate Figure 26, which presents the Accuracy of the interference estimation algorithm.

Table C-3. Data for "accuracy of interference estimation algorithm"

Packet rate (pkt/sec)	Accuracy	deviation
1	0.9401	0.024
2	0.9328	0.02466
3	0.9244	0.03014
4	0.9267	0.03227
5	0.8971	0.06148
6	0.8863	0.07573
7	0.88	0.07679
8	0.8494	0.10942
9	0.8403	0.11305

Table C-4 shows the data that we use to generate Figure 27 (a), which presents the detection rate for our detection on DoQoS attacks.

Table C-4. Data for "detection rate of DoQoS attacks"

Packet rate (pkt/sec)	Detection rate	deviation
1	0.82647	0.07443
2	0.81949	0.07091
3	0.81386	0.07325
4	0.81176	0.07267
5	0.79733	0.08001
6	0.79721	0.07748
7	0.79705	0.07202
8	0.77867	0.06959
9	0.76029	0.06783

Table C-5 shows the data that we use to generate Figure 27 (b), which presents the detection rate for the detection on QoS attacks.

Table C-5. Data for "detection rate of QoS attacks"

Packet rate (pkt/sec)	Detection rate	deviation
1	0.73213	0.023
2	0.7359	0.0281
3	0.73001	0.0296
4	0.72624	0.0303
5	0.72118	0.0299
6	0.72127	0.0336
7	0.69465	0.0439
8	0.68597	0.0586
9	0.66209	0.0607

Table C-6 shows the data that we use to generate Figure 28, which presents the false positive rate of detection on DoQoS attacks.

Table C-6. Data for "false positive rate of detection on DoQoS attacks"

Packet rate (pkt/sec)	False positive rate	Deviation
1	0.26031	0.0177
2	0.25995	0.0152
3	0.25967	0.0168
4	0.24654	0.017
5	0.24603	0.0168
6	0.24607	0.0154
7	0.24062	0.0159
8	0.22587	0.0157
9	0.22355	0.0146

VITA

NAME: Bin Lu **ADDRESS**: 301 H.R.Bright Building
EMAIL: binlu@tamu.edu College Station, TX 77843-3112

EDUCATION

Ph.D. in Computer Science, Texas A&M University, August 2005
M.S. in Computer Science, Harbin Institute of Technology, PR China, July 1998
B.S. in Computer Science, Harbin Institute of Technology, PR China, July 1996

EXPERIENCE

Teaching Assistant, Department of Computer Science, Texas A&M University, 2002-2005

Research Assistant, Department of Computer Science, Texas A&M University, 1999-2002

Graduate Assistant, Department of Computer Science, Texas A&M University, 1998-1999

PUBLICATIONS

Bin Lu and Udo W. Pooch, "Security in QoS Signaling Systems for Mobile Ad Hoc Networks," to appear in proceedings of *the 2005 IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob'05),* Montreal, Canada, August 22-24, 2005.

Bin Lu and Udo W. Pooch, "A Lightweight Authentication Protocol for Mobile Ad Hoc Networks," in proceedings of *the IEEE International Conference on Information Technology: Wireless Ad Hoc/Sensor Networks and Network Security (ITCC 2005),* Las Vegas, NV, April 2005, pp. 546 – 551.

Bin Lu and Udo W. Pooch, **"**A Game Theoretic Framework for Bandwidth Reservation in Mobile Ad Hoc Networks," in proceedings of *the IEEE International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks (QShine 2004),* Dallas, TX, October 2004, pp. 234-241.

Bin Lu and Udo W. Pooch, "Security in QoS Models and Signaling Systems for Mobile Ad Hoc Networks," in proceedings of *the 2003 International Conference on Wireless Networks (ICWN 2003),* Las Vegas, NV, June 2003, pp. 563-569.

Bin Lu and Udo W. Pooch, "Cooperative Security-Enforcement Routing in Mobile Ad Hoc Networks," in proceedings of *the 4th IEEE International Conference on Mobile and Wireless Communications Network (MWCN 2002),* Stockholm, Sweden, September 2002, pp.157 – 161.